

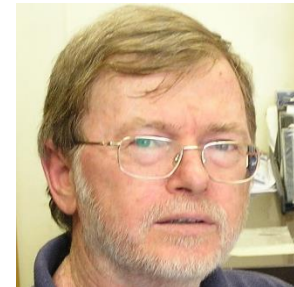
# Model Based Continuous ~~Systems~~ Engineering

How to Reduce Overhead and  
Increase the Value of MBE



**Michael Masin**  
*Systems & IoT Engineering*

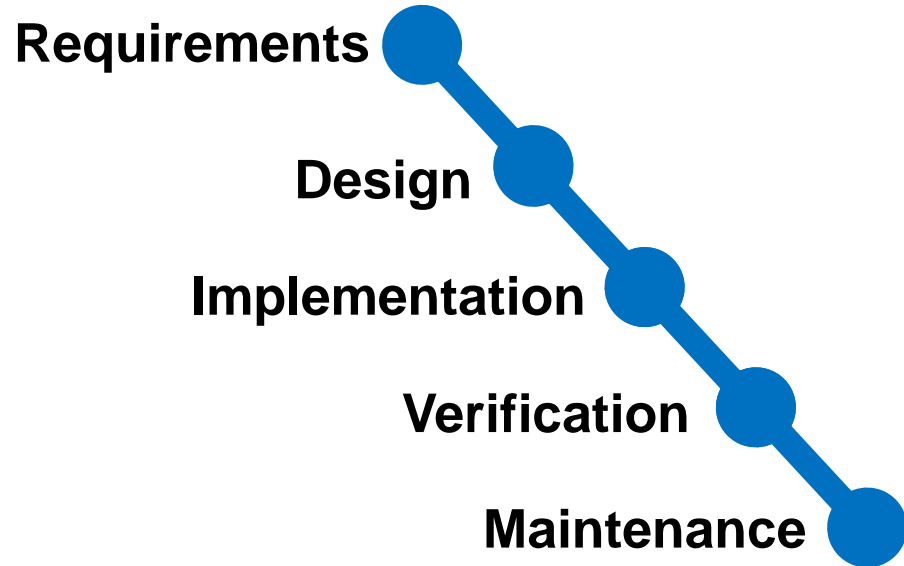
# Thanks to *Systems & IoT Engineering* group (... and others)



# Agenda

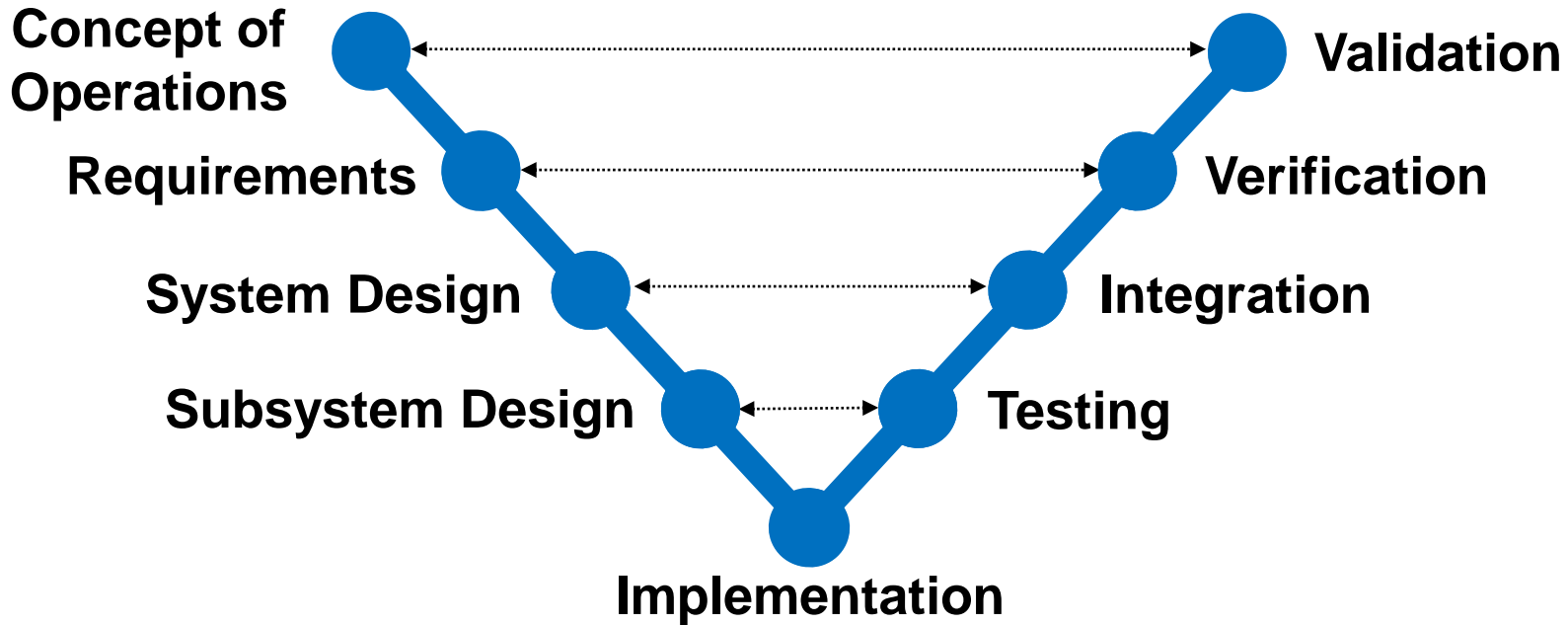
- **From SE to MBCE**
- AOW background
- PORTALS
- FAME
- EMI
- DANSE
- Summary

# Waterfall Model (Benington 1956, Royce 1970)



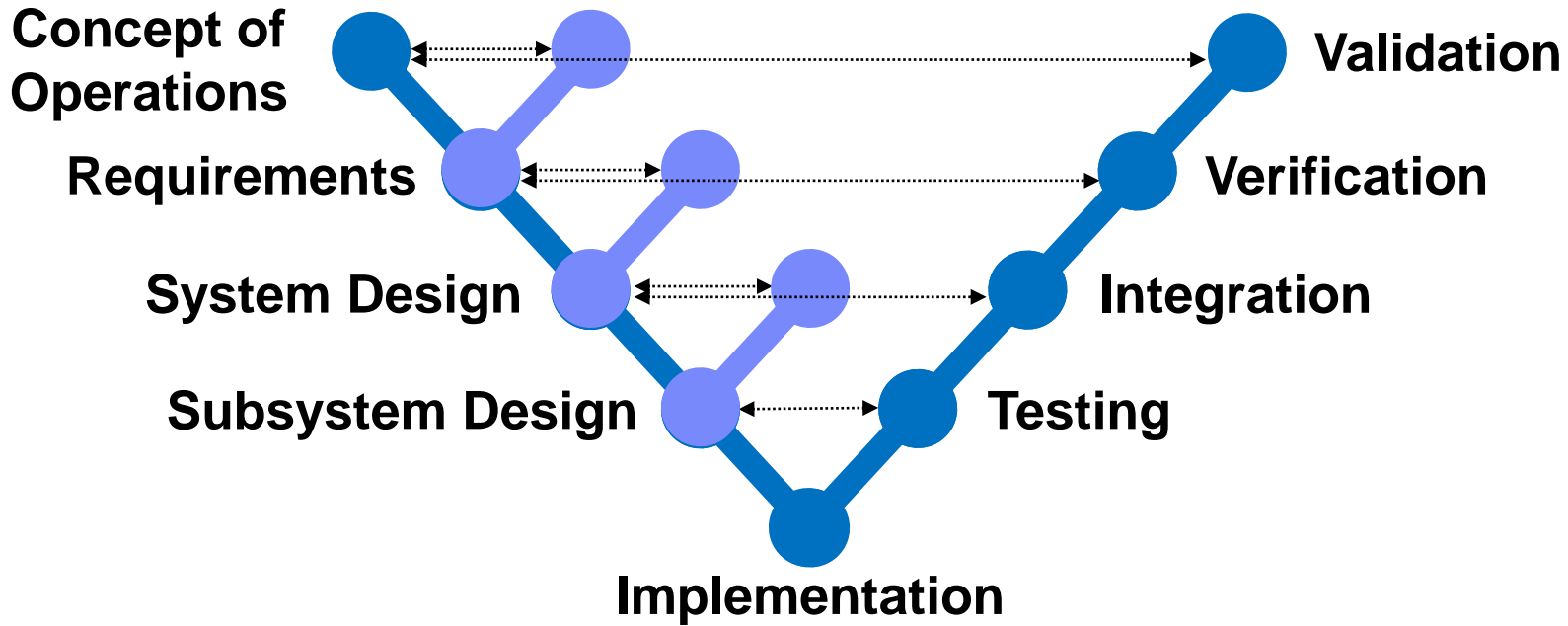
**Start of SE**

# V Model (~1980s)



**SE as we know today**

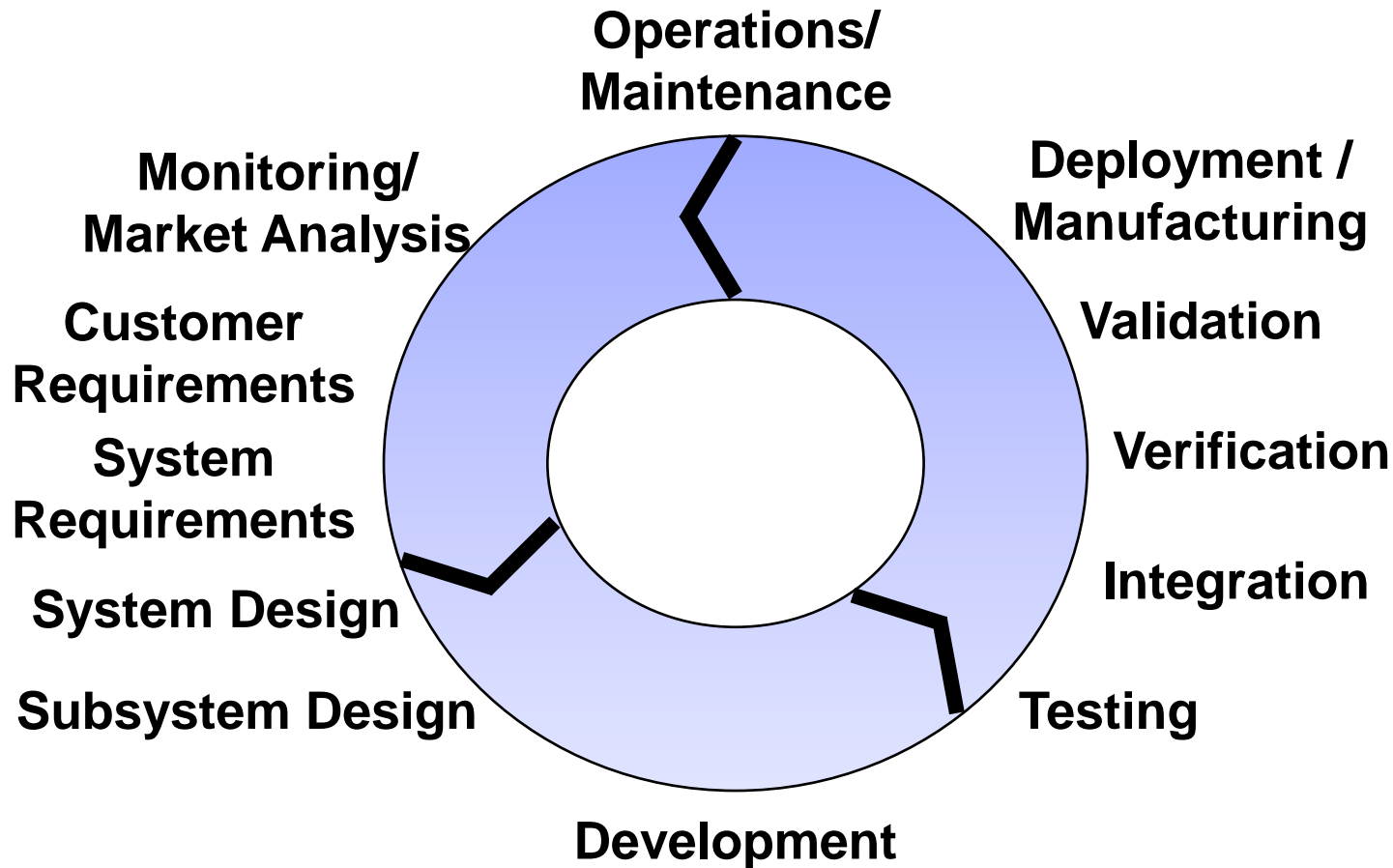
# W Model (recent 5-10 years)



 Virtual models

**MBSE**

# Model Based Continuous Engineering (MBCE)



**Use models to create models  
Virtual / real environment**

# Agenda

- From SE to MBCE
- **AOW background**
- PORTALS
- FAME
- EMI
- DANSE
- Summary

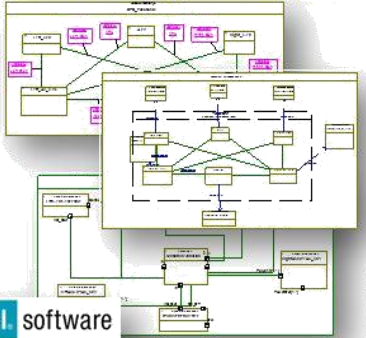


# Architectural Design

- System Complexity is increasing – manual decisions no longer possible
- Solution approaches:
  - Levels of Abstraction
    - Mapping from Layer  $i$  (Requirements) to Layer  $i+1$  (Architecture)
  - Separation of concerns
    - Multiple viewpoints: functional, technical, geometrical, safety, timing, ...
    - Modeling Viewpoints vs. Analysis Viewpoints
    - Independent asynchronous development
  - Tools
    - Modeling
      - Component Based Design
    - Analysis
      - Domain specific tools
      - Extension of modeling tools
      - Black box integration
      - Custom optimization modeling

# IBM's Architecture Optimization Workbench Concept

1. Describe system through different SysML views, including design alternatives, constraints and goals



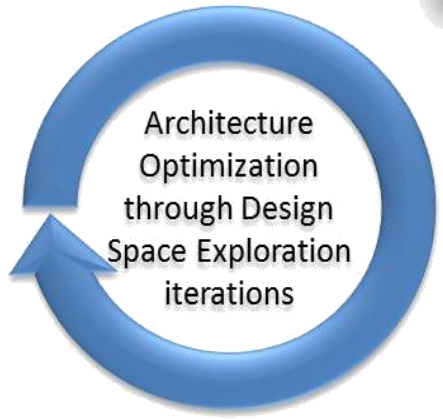
Rational software Rhapsody

2. Derived Data Schema for Input and output structures

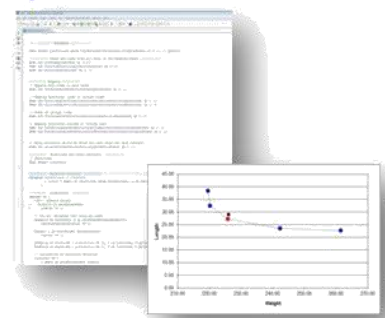
id	name	guid	variants
1	Junction	GUID e1eb1581-c487-4be2-9fd6-8ce56a53b2ea	Cat Junction
3	Relay	GUID 8feb223d-5bc6-40d6-b179-46b5ff3d7e58	Cat Relay

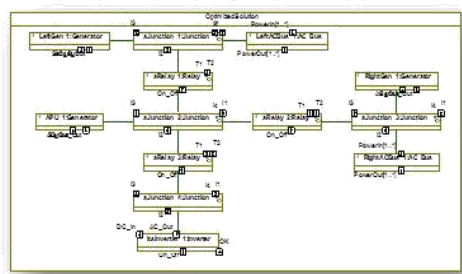
id	name	guid	typed type name	variants	base from to
23	Pacemaker	GUID 39e128-9e09-4901-493a-e834e34332e2	23 Pacemaker	no Variants	23 0 100
10	Relay	GUID 8feb223d-5bc6-40d6-b179-46b5ff3d7e58	10 Relay	no Variants	10 0 100
23	Pacemaker	GUID 80d39f-030e-4d08-d743-d0e75687336c	23 Pacemaker	no Variants	23 0 100
13	Relay	GUID 80442e-984e-4977-40a3-c09f91686e97	13 Relay	no Variants	13 0 100
20	Relay	GUID 3421210a-79e4-4232-897c-b0b552330383	20 Relay	no Variants	20 0 100
31	Relay	GUID 33b086-3309-4c0e-e82e-79d3fae0a042	31 Relay	no Variants	31 0 100
24	Relay	GUID 2120e631-ea08-477b-c07e-244e831a4724	24 Relay	no Variants	24 0 100
21	Relay	GUID 3a027020-8914-4a0e-810d-c9211e631d42	21 Relay	no Variants	21 0 100
11	Relay	GUID 88913a79-95d-44d8-377e-1852-6651d12	11 Relay	no Variants	11 0 100
14	Relay	GUID 860a77-0068-44d0-0322-872243e28d9f	14 Relay	no Variants	14 0 100
22	Relay	GUID ac7b3a7031-400e-8636-d81d1c1c829e	22 Relay	no Variants	22 0 100



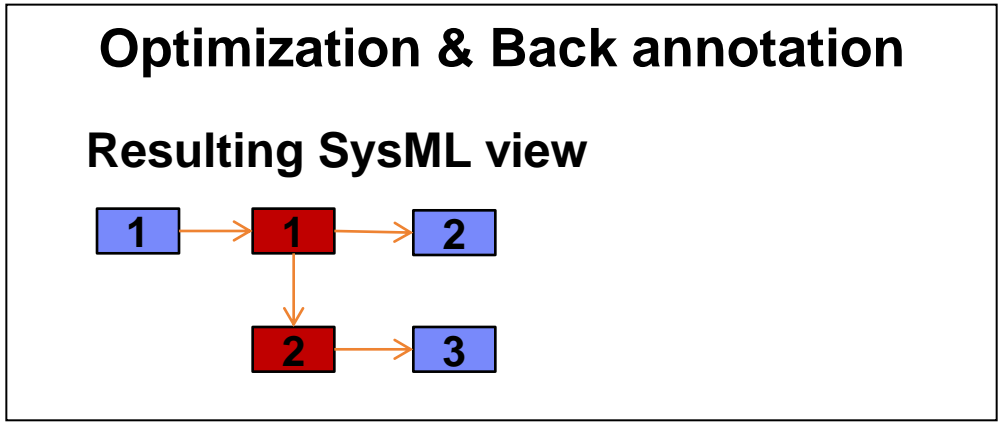
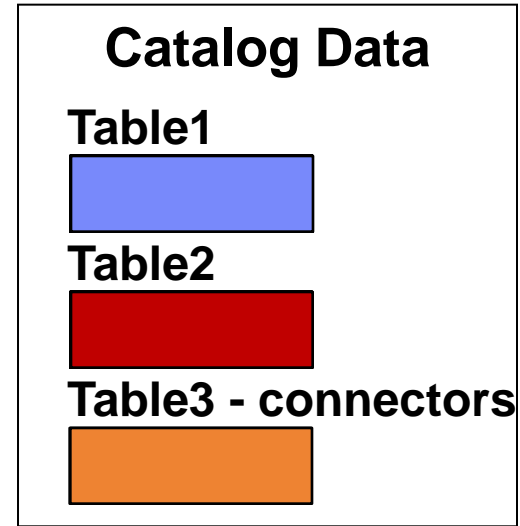
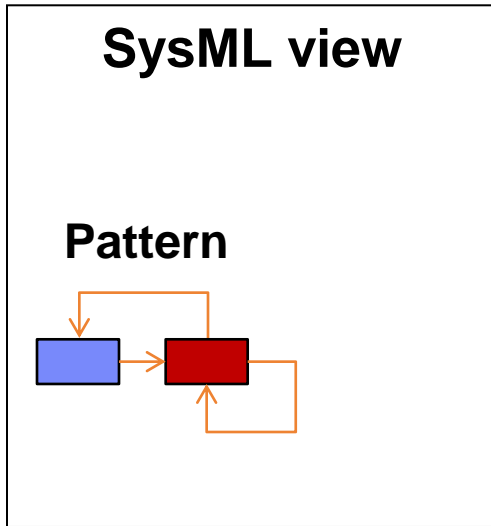
3. Automatic translation (via an interchange format) into Optimization solver



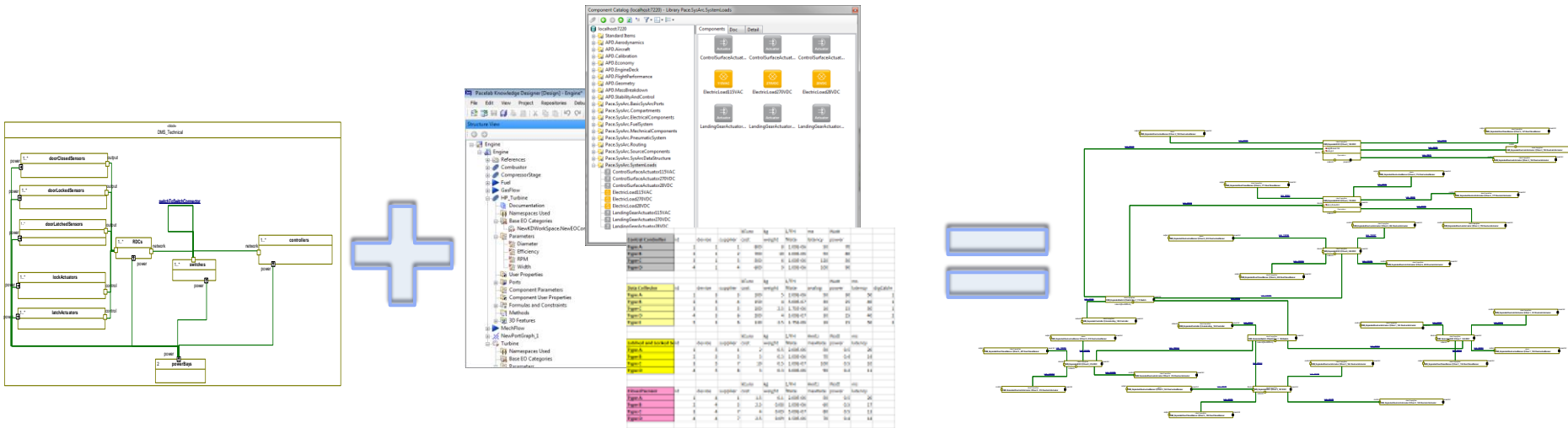
4. Optimized architecture back annotated to SysML model



# Concise modeling = SysML views + Data



# AOW uses Concise Modeling



- **Large systems architectures are difficult to model**
  - Lots of elements and details
  - Time consuming
  - Error prone
- **Concise Modeling – SysML models combined with tabular data**
  - SysML depicts the system composition rules
  - Tables contain instantiations (catalogs of types and inventories of parts)
- **Component libraries**
- **Optimization fills missing attribute values or inventory tables**

# Semantic Middleware (SEMI)

- **Paradigm shift:** from *Classification-by-Containment* to **Classification-by-Property**
  - suggested by Parsons and Wand (2000) for information management
  - define *things* that possess properties
    - hierarchy of properties
  - no a-priory classification, open world assumption
  - *classes* are defined by set of properties
    - things could belong to many classes
  - *instance, class and property bases*
- **Main API: AttributeSet(A)**, where A is a set of attributes
  - returns instances that have all attributes in set A
  - e.g., **AttributeSet**({Cost}) returns all instances that have attribute “Cost”
- **Domain specific ontologies**
- **Design Space Exploration (DSE) ontology**
  - Attributes could be either variables or parameters
  - *isSelected*

# Pluggable Analysis Viewpoints

- from

*totalCost* =

$$\sum_{j \in \text{SensorTypes}} \sum_{i \in \text{Sensors}} \text{SensorType}[j].\text{Cost} \cdot \text{sensor}[i][j] + \dots$$

$$+ \sum_{j \in \text{SwitchTypes}} \sum_{i \in \text{Switches}} \text{SwitchType}[j].\text{Cost} \cdot \text{switch}[i][j]$$

- to

$$\text{totalCost}(i) = \sum_{j \in \text{AttributeSet}(\{\text{Cost}\})} \text{isSelected}(j) \cdot \text{Cost}(j) \quad \forall i \in \text{AttributeSet}(\text{totalCost})$$

# AOW allows for Multi Objective Optimization

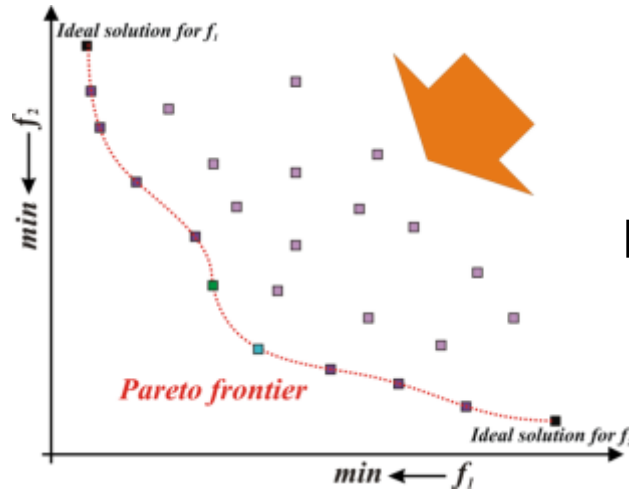
## Optimization Problem Formulation (CPLEX)

```

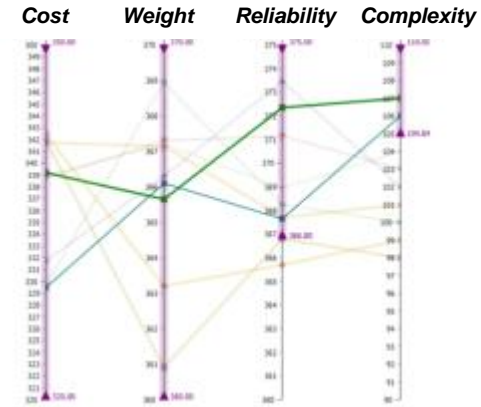
145 minimize objvalue * totalWeight + lengthGravity * totalLength
146
147
148 ##### Constraints #####
149
150 subject to
151
152 // Constraint to use Plane 20, e.g., according to the System Exp suggestion for a given weight relation
153 // UsePlanes
154 // Force() in plane() : 1:0 == 1: usePlane() == 1
155
156 ##### Objective section #####
157
158 TotalObjConst
159 totalLength == sum() in plane, p in pPlane pobjplane[p] * p.weight
160 // sum() in plane, w in wires wireplane[p] * w.weight * w.length
161 // sum() in load, j in plane, l in loadplane j.load * l.d at l.pPlane == 1:0, w in wires
162 // wireplane[p][j][l] * w.weight * w.length * l.p.d at l
163
164 TotalLengthConst
165 totalLength == sum() in plane usePlane[] * j.length
166 // sum() in load, j in plane, l in loadplane j.load * l.d at l.pPlane == 1:0
167 // planeLoad[][] = l.p.d at l
168
169 ##### Shipping contracts #####
170
171 // Generalized to functional ipm planeholder per functional link
172 // usePlaneConstraints
173 // Force() in load, sum() in plane usePlane[] == 1
174
175 // Technical to generalised ipm to pPlane usePlane[]
176 // usePlaneConstraints
177 // Force() in plane, sum() in pPlane pobjplane[p] == usePlane[]
178
179 // Technical to generalised wire to pPlane link
180 // usePlaneConstraints
181 // Force() in plane, sum() in wire wireplane[p] == usePlane[]
182
183 // Technical to generalised wire to load link
184 // usePlaneConstraints
185 // Force() in load, j in plane, wire in wire wireplane[j][l] == planeLoad[]
186
187 ##### Network constraints #####
188
189 // If pPlane usePlane is used for any load, it should be used
190 // usePlaneConstraints
191 // Force() in load, j in plane usePlane[] == planeLoad[]
192
193 // If pPlane usePlane is NOT used for any load, it should NOT be used
194 // usePlaneConstraints
195 // Force() in plane usePlane[] == sum() in load planeLoad[]
196
197
  
```



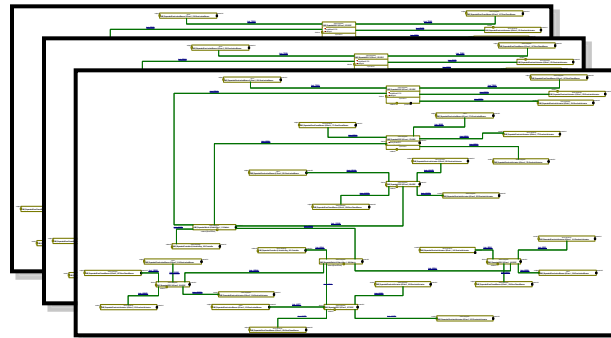
## Calculation of Efficient Frontier



## Multi Objective visualization



## Architectures on efficient frontier



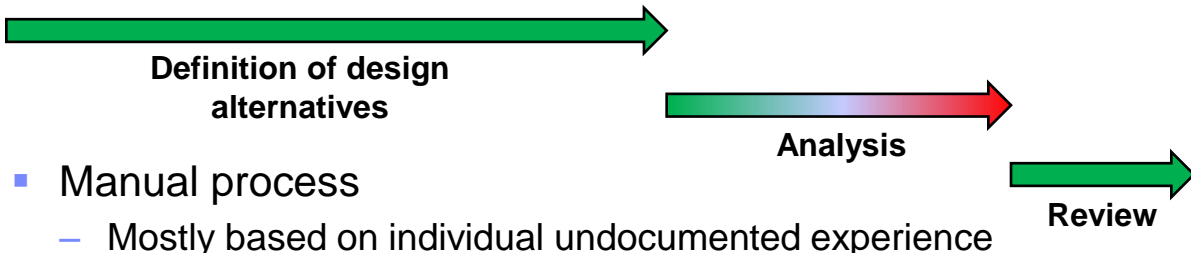
## Selection of desired frontier



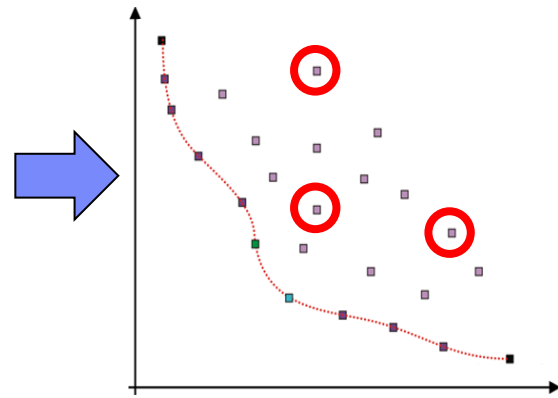
Next iteration

# AOW Vision: Accelerate Smarter Architectural Decisions

NOW

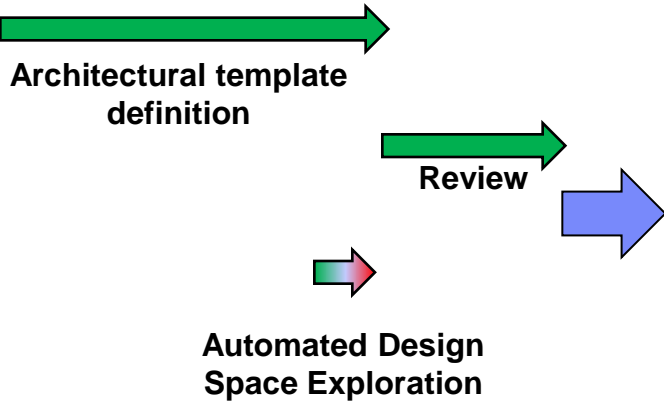


- Manual process
  - Mostly based on individual undocumented experience
  - Lots of reuse of previous designs
  - No assurance for solution optimality
  - Each additional design – linear effort
- No formal capture of designs
  - Modeling for specific analysis types

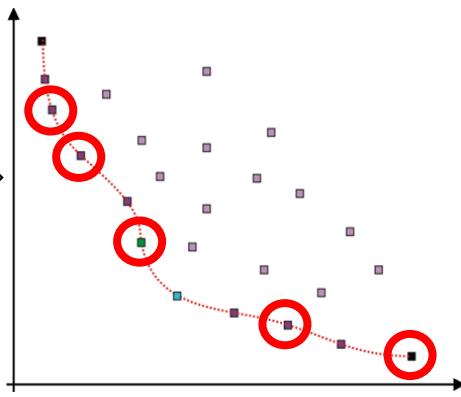


time

With AOW



- Faster design
- Smarter designs
- Transparency, traceability
- Knowledge capture and reuse
- Optimization Capabilities in the hands of Architects





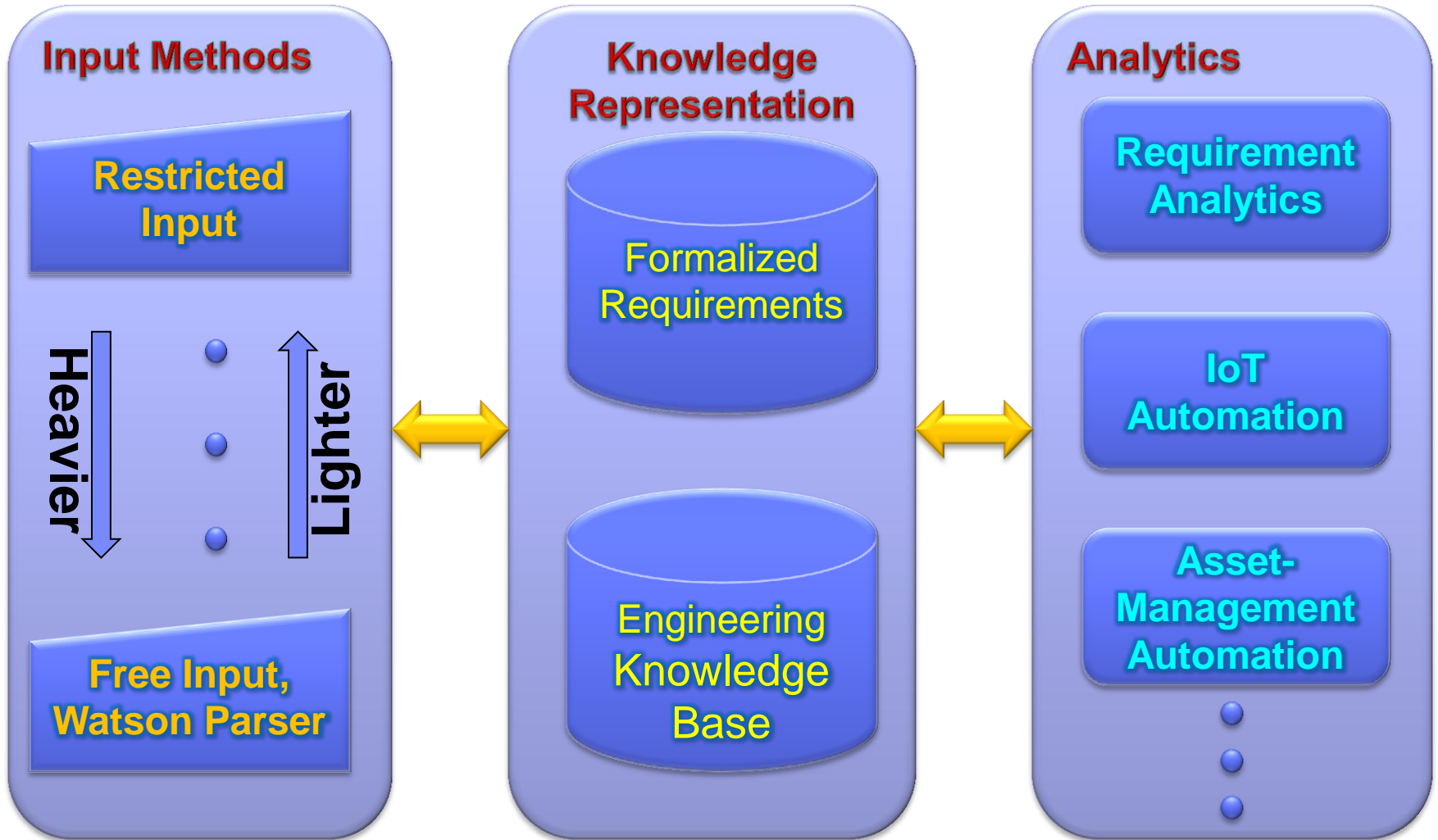
# Agenda

- From SE to MBCE
- AOW background
- **PORTALS**
- FAME
- EMI
- DANSE
- Summary

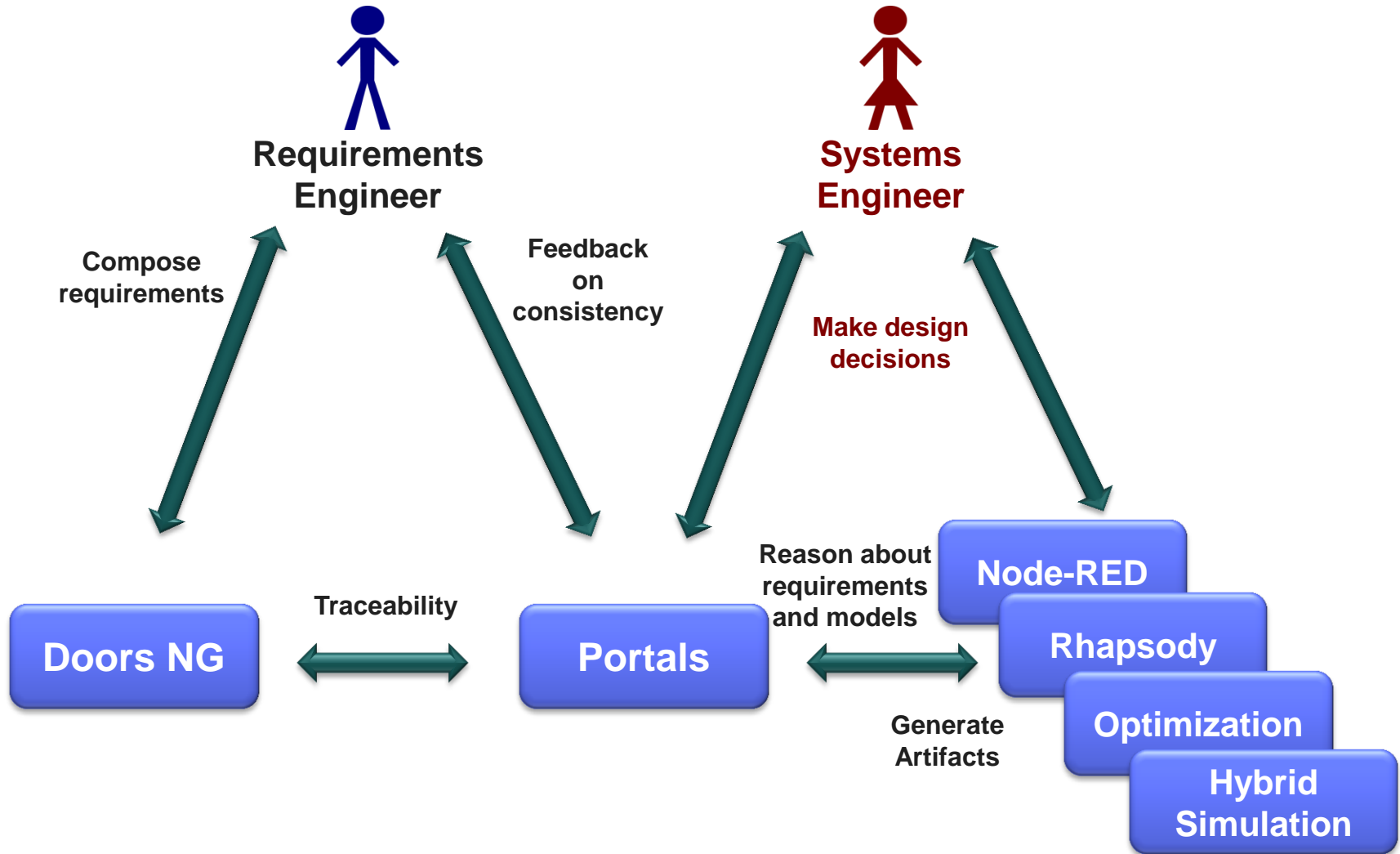
# PORTALS

- PORTALS is a research project whose goal is
  - to create tools to assist requirements engineers in incrementally raising the formalization level of system requirements, and
  - to use formalized requirements to
    - provide feedback on the quality of the requirements (e.g., identifying omissions and contradictions), and
    - create downstream artifacts (e.g., models, monitors, tests, code)

# PORTALS Architecture

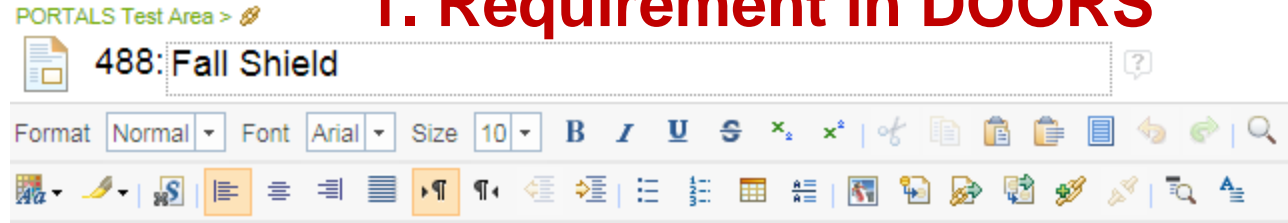


# Personas and interactions

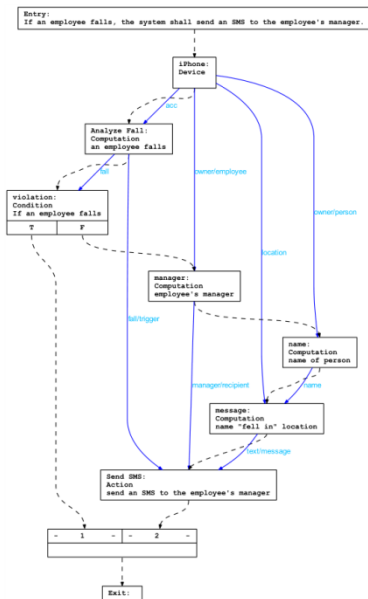


# Scenario 1: WorkRight Fall Shield

## 1. Requirement in DOORS



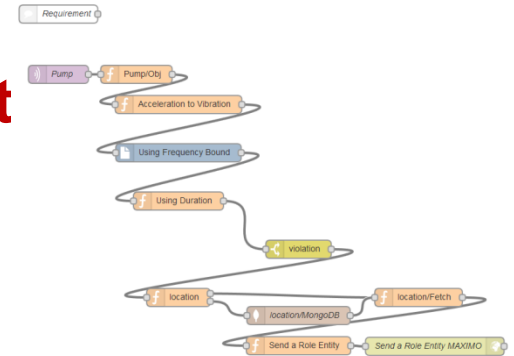
## 3. Process Model



## 2. Paraphrase by PORTALS

if "an employee" falls then "the system" shall send [an abstract entity] "an SMS" (direction) "manager" of "the employee's"

## 4. Implementat in Node-RED



# Scenario 2: IoT Pump

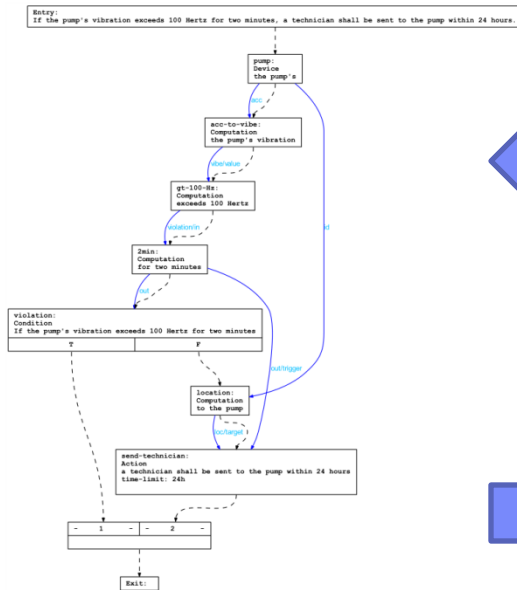
## 1. Requirement in DOORS

PORTALS Test Area > 487: Too much vibration

Format Normal Font Arial Size 10 B I U

If the pump's vibration exceeds 100 Hertz for two minutes, a technician shall be sent to the pump within 24 hours.

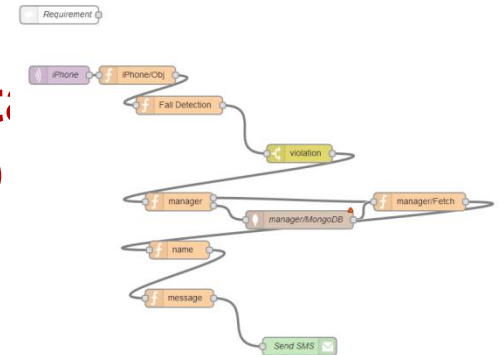
## 3. Process Model



## 2. Paraphrase by PORTALS

if "vibration" of "the pump's" is greater than 100 Hz (duration) 2 min then "?" shall send [a role entity] "a technician" (duration) 24 hr; (direction) "the pump"

## 4. Implement in Node-RED



# Engineering Knowledge Base: WorkRight Catalog

## Events

### Overexertion



Heart Rate



Body Temperature



Sweat & Tears



### Fall Detection



Accelerometer



Radar



## Devices / Systems

### iPhone

Accelerometer

Orientation

Vibration

Send SMS

Device ID

Owner ID

### Fever Smart

Temperature

Device ID

### TI SensorTag

Accelerometer

Temperature

Device ID

## Actions

Send SMS



Send Mail



Vibrate



## Services

Employee To Manager



Device ID To Person



# Engineering Knowledge Base: IoT Catalog

## Events

### Check Bound

Electric Threshold

Duration

Frequency

### Check Range

Electric Threshold

Duration

Frequency

## Devices / Systems

### Pump

 Accelerometer

 Location

 Device ID

## Actions

Send SMS



Send Role



Send Mail



## Services

Device ID To Location



Acceleration To Vibration





# Agenda

- From SE to MBCE
- AOW background
- PORTALS
- **FAME**
- EMI
- DANSE
- Summary

# FAME: Framework for Affordability Modeling & Evaluation

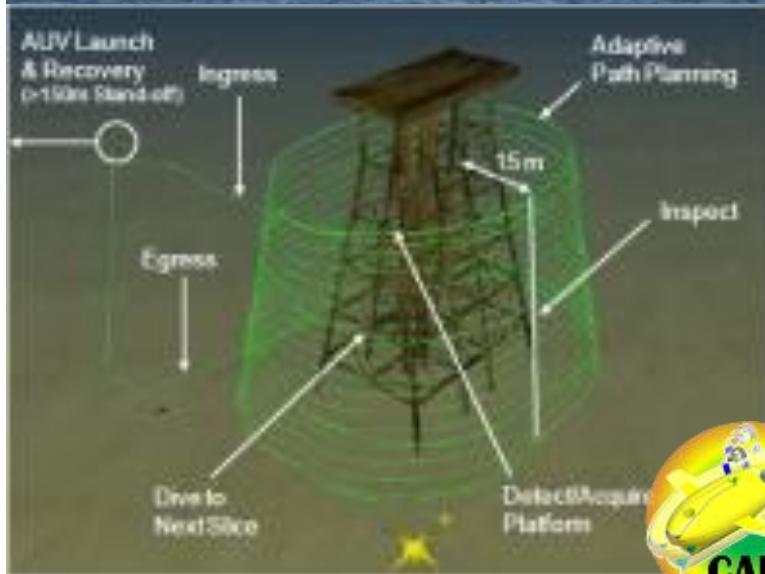
- Joint project with Lockheed Martin
- Currently: Tools exist for product optimization and operations optimization with no links in between
- Goal: Reduce operating costs by **optimizing the product** (and not the operations)



## Solution:

- Use design models to find optimal operation strategies (e.g. maintenance)
- Use design models for asset management tools customization

# FAME Use Case: Unmanned Underwater Vehicle



- Metrics
  - Production costs (up-front materials & fabrication)
  - Mission costs (vessel & crew for vehicle deployment & recovery)
  - Maintenance costs (planned upkeep & corrective repairs)
  - Availability (is the vehicle operational when it is needed)
- UUV designs are valued using LM Palm Beach **CAUSE** analysis tool (**C**onceptual **A**utomated **U**UV **S**izing **E**nvironment)

**Selection of Vehicle Architecture  
Directly Influences Cost and  
Availability Metrics**

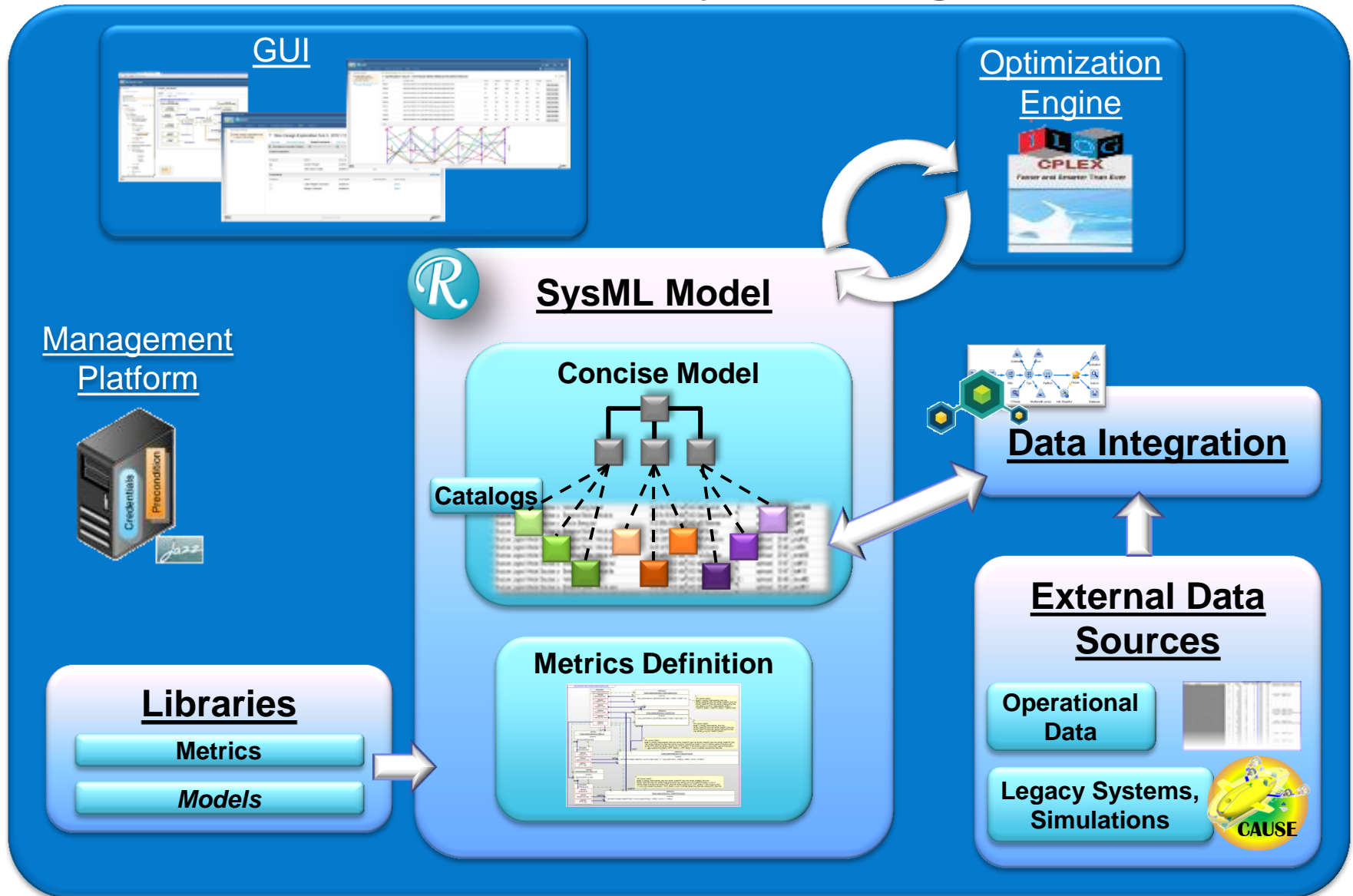
## FAME New Capabilities

- Definition of libraries to capture reusable common metrics and processes
  - Key metrics: life cycle cost & system availability
- Expansion of user viewpoints feeding optimization
  - External legacy analysis models
    - Ability to protect IP of source models through “Black Box” integration
  - Existing operational data from procurement, fabrication, and operations & maintenance systems

**FAME Enhances Reuse and Usability Features of AOW**

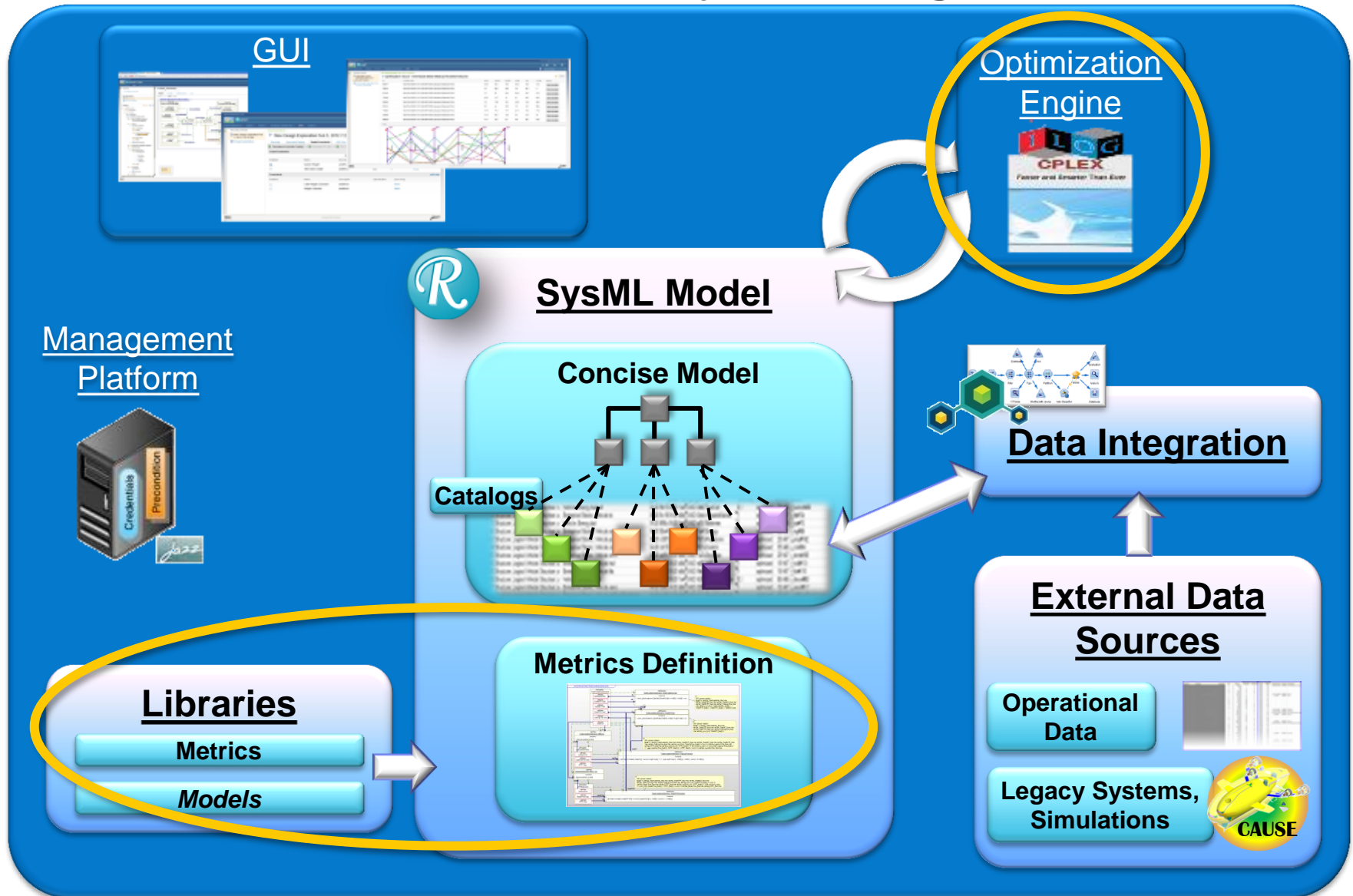
# FAME

## Framework for Affordability Modeling & Evaluation

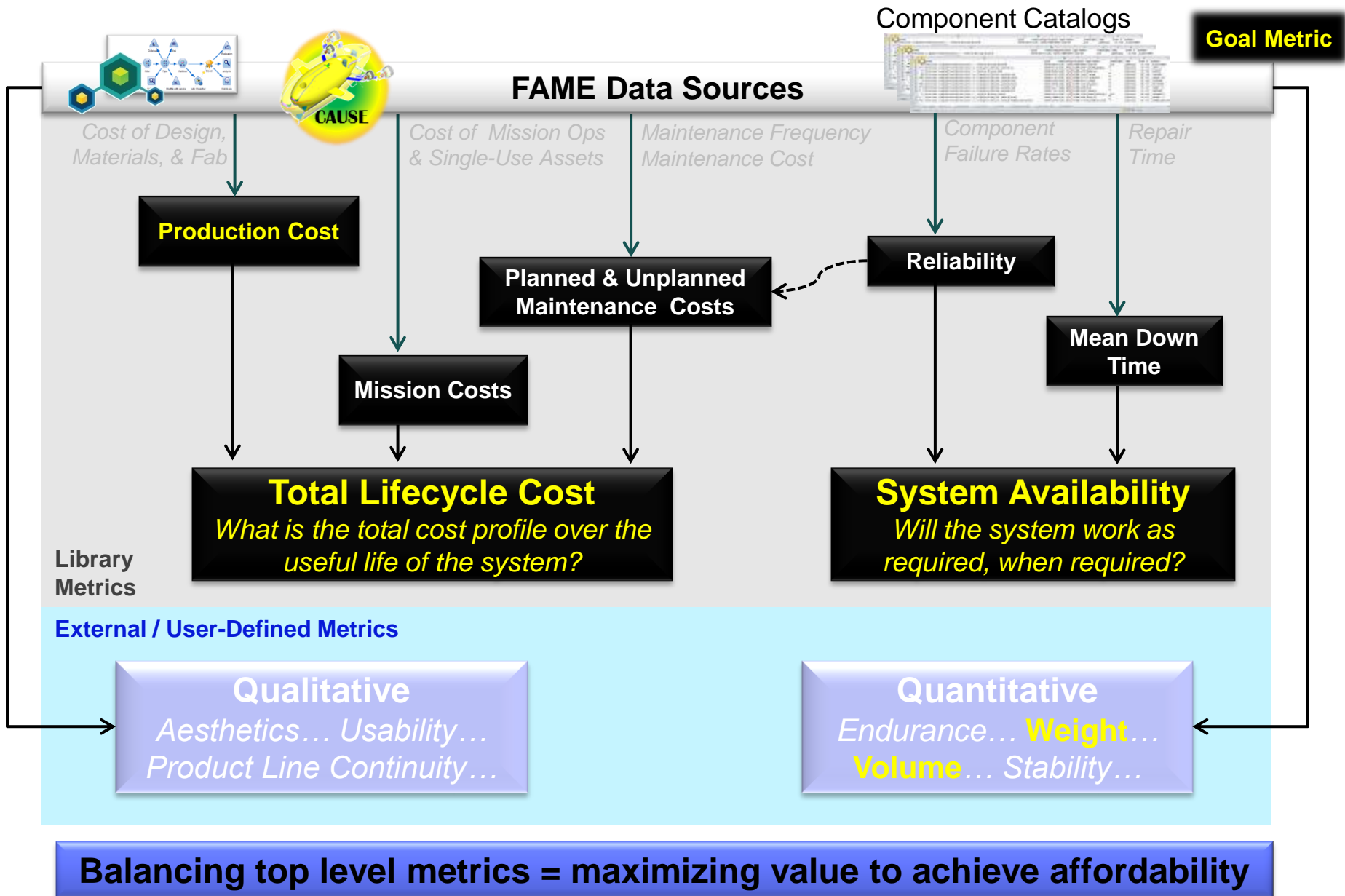


# FAME

## Framework for Affordability Modeling & Evaluation



# Metrics for Affordability



# Metrics Library Concept

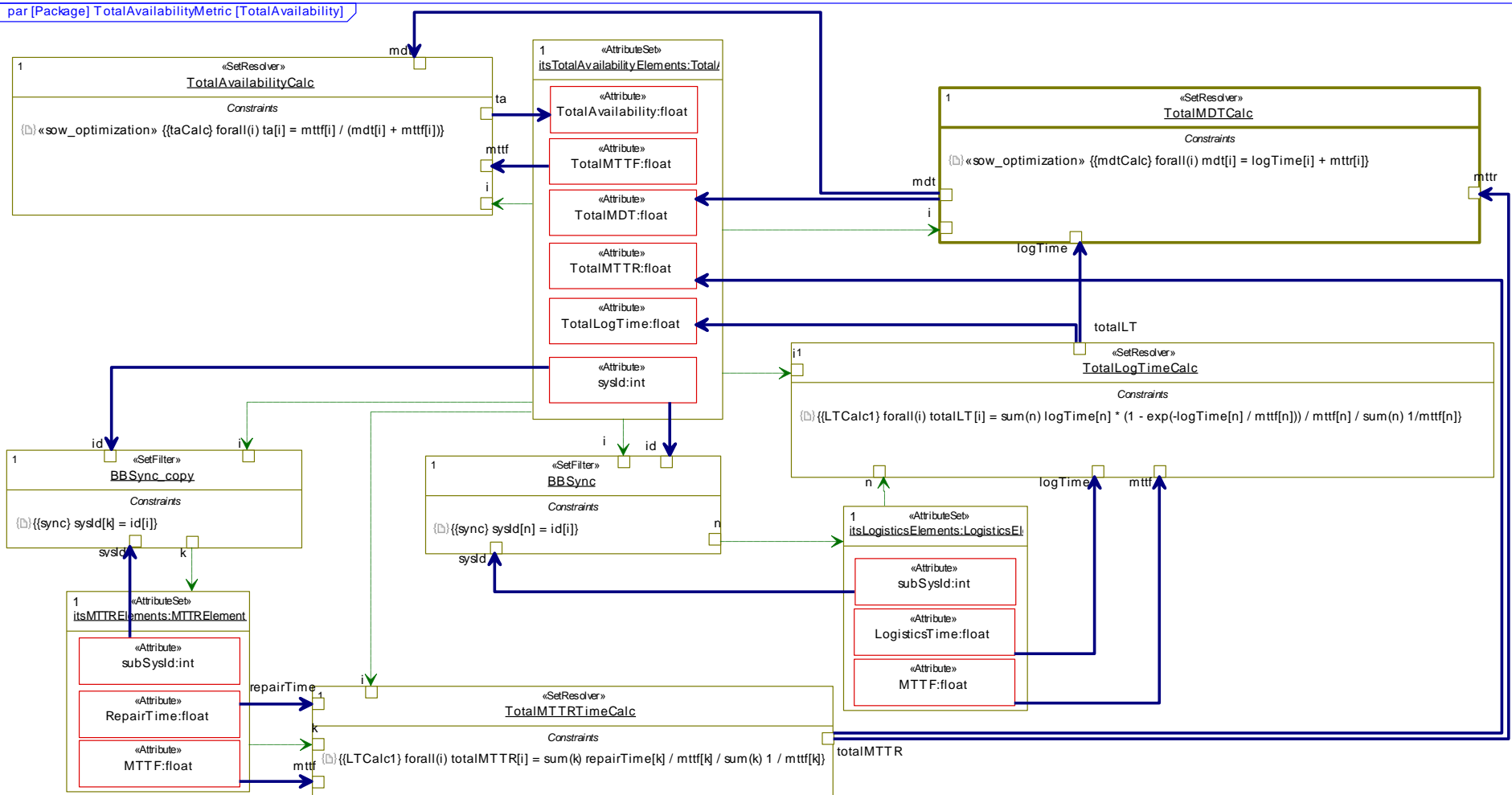
- SysML Parametric Diagram notation captures the attributes and formulas that define Goal Metrics for the optimization
- Blocks define sets of attributes required of architecture elements to support computing the given metrics
  - Block elements are classification-by-property attribute sets
  - Architecture elements can inherit attribute sets

**Common Library Metrics Are Independent of a Particular Design or Domain Model**



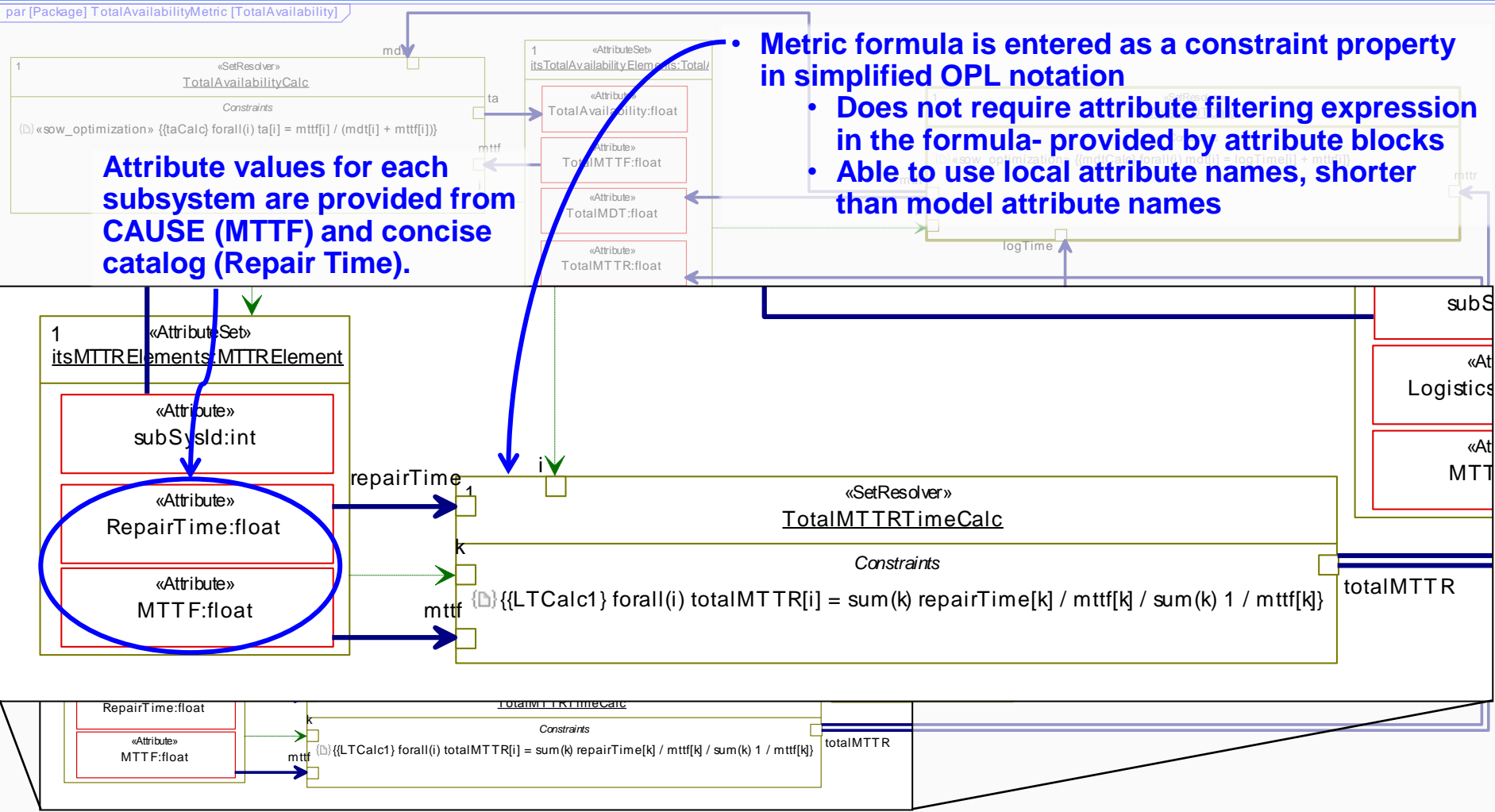
# FAME Metrics Library

## System Availability



# FAME Metrics Library

## System Availability

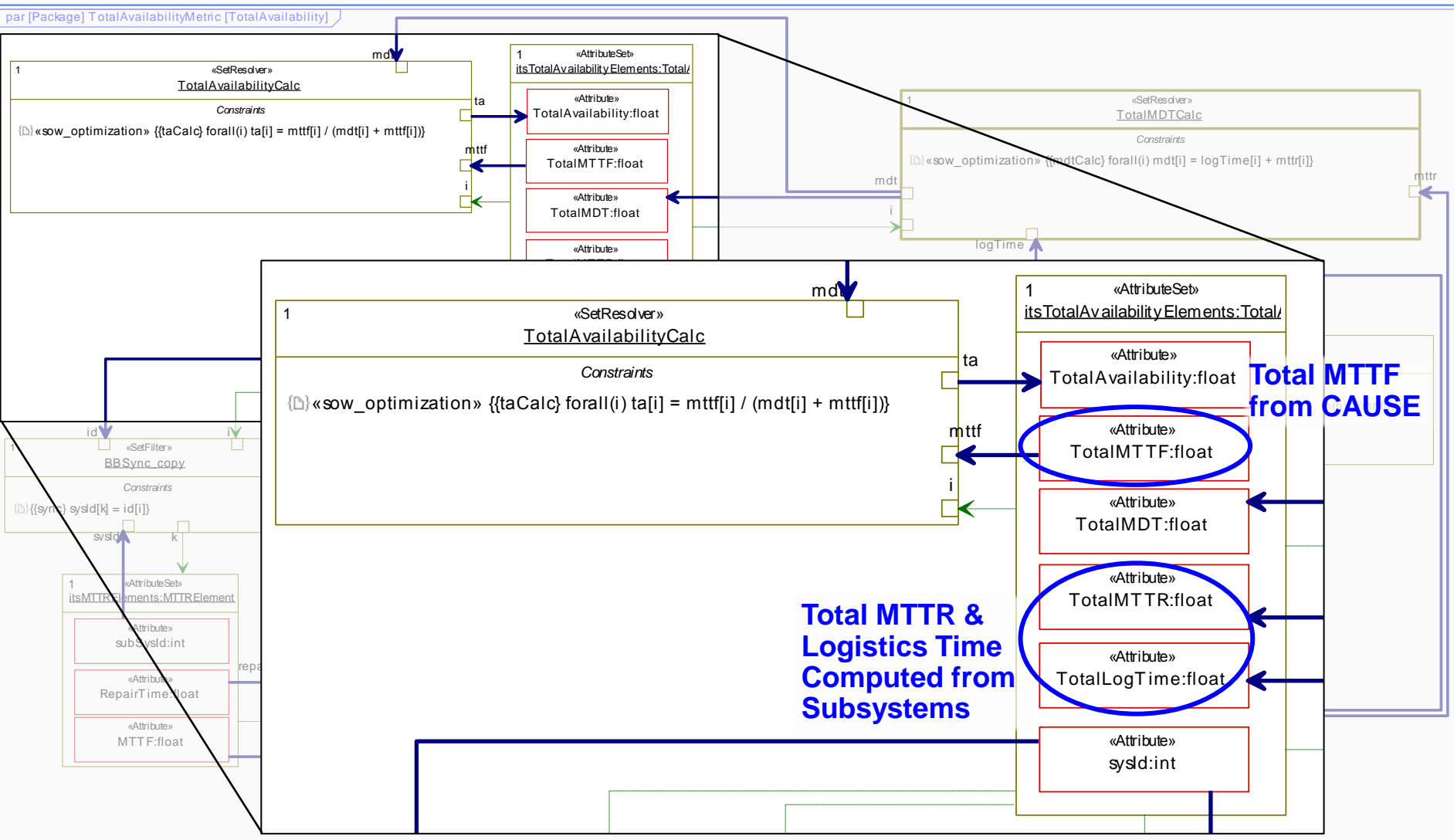


Attribute values for each subsystem are provided from CAUSE (MTTF) and concise catalog (Repair Time).

- Metric formula is entered as a constraint property in simplified OPL notation
- Does not require attribute filtering expression in the formula- provided by attribute blocks
- Able to use local attribute names, shorter than model attribute names

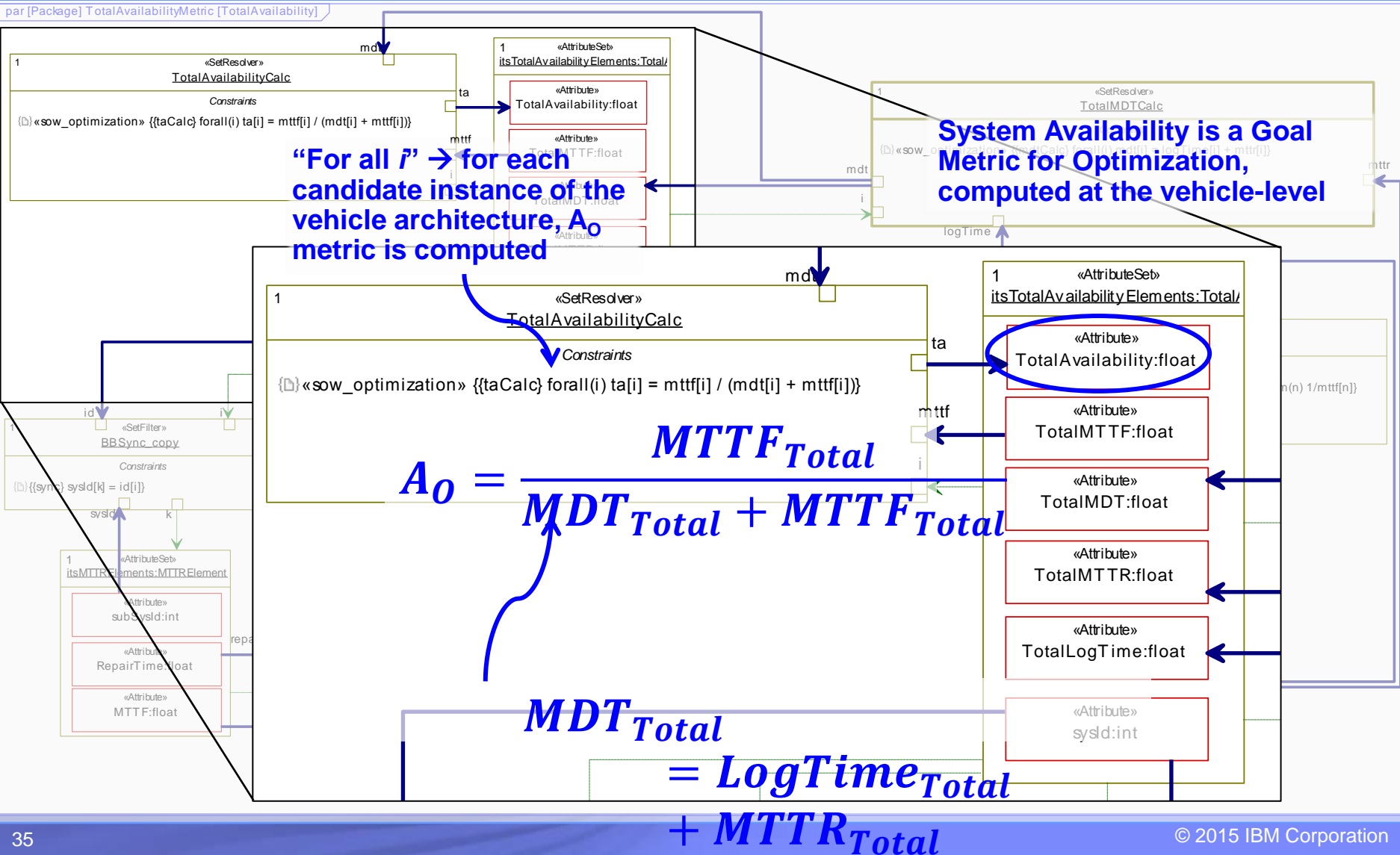
# FAME Metrics Library

## System Availability



# FAME Metrics Library

## System Availability



“For all  $i$ ”  $\rightarrow$  for each candidate instance of the vehicle architecture,  $A_0$  metric is computed

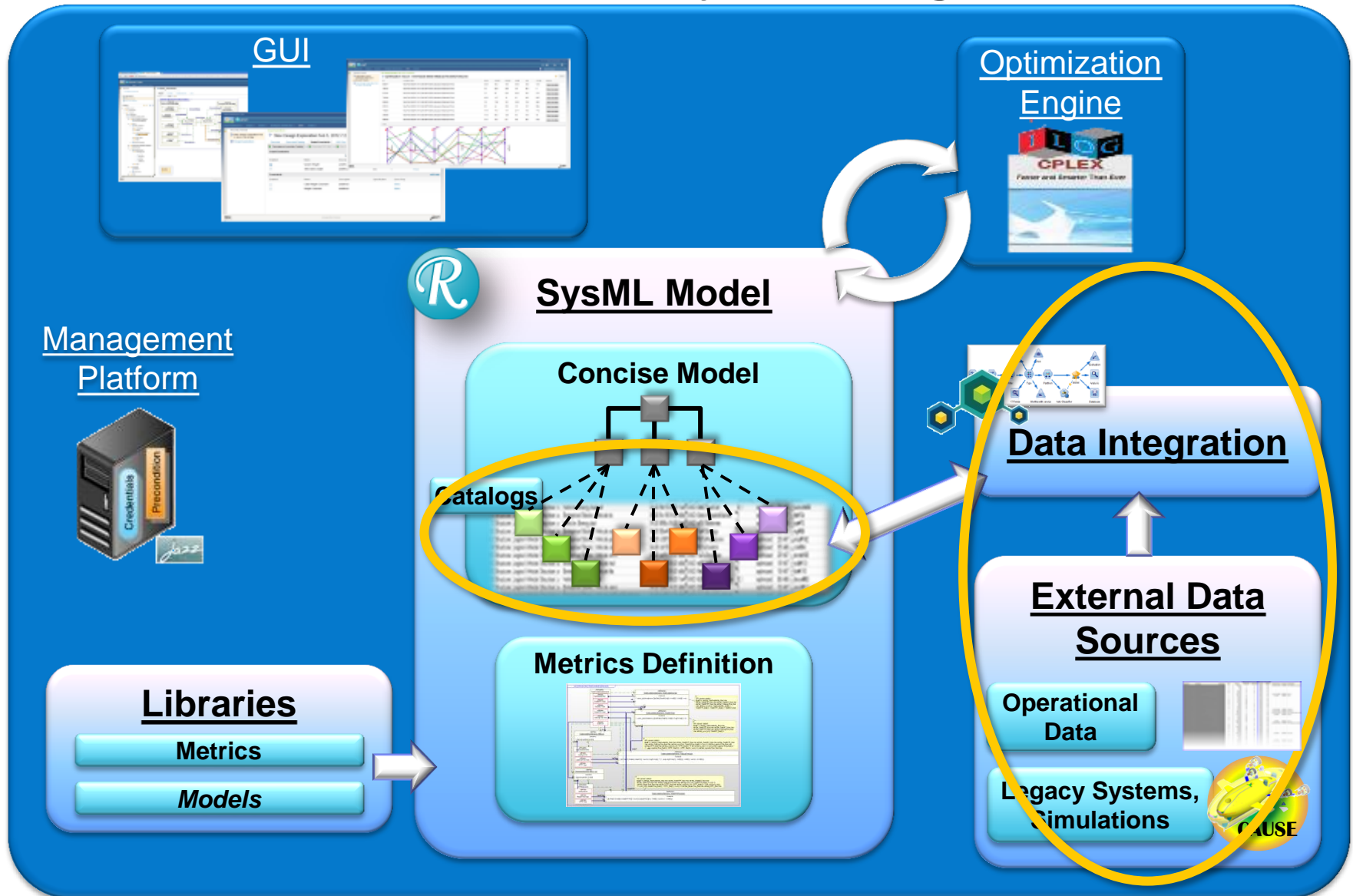
System Availability is a Goal Metric for Optimization, computed at the vehicle-level

$$A_0 = \frac{MTTF_{Total}}{MDT_{Total} + MTTR_{Total}}$$

$$MDT_{Total} = \text{LogTime}_{Total} + MTTR_{Total}$$

# FRAME

## Framework for Affordability Modeling & Evaluation



# Catalog Data

- Subsystem catalogs include key parameters differentiating physical options for each subsystem
  - Reliability Parameters
    - Maintenance Frequency
    - Maintenance Duration
    - Redundancy
  - Cost Parameters
    - Per-Mission Operations Cost

## Energy Catalog

int	int	int	int	float	float	float
id	energy_type	IN_ENERGY_specificE	energy_fuel_cell_type	MissionCost	MaintenanceFreq	MaintenancePlanCost
3500000	1	68	1	0	18	600
3500000	1	39	1	0	18	1000
3500000	1	62	1	0	18	400
3500000	1	69	1	0	18	800
3500000	2	68	1	0	18	1200
3500000	2	68	1	0	18	800
3500000	2	68	1	0	18	1200
3500000	2	68	1	0	18	800

## Propulsion Catalog

int	float	float	float	float	float
id	IN_SUB_propulsor_type	IN_SUB_propulsor_ducted	IN_VEH_propulsors_num	IN_SUB_propulsor_redundancy	MaintenanceFreq
13500000	1	0	1	0	35
13500001	1	1	1	0	35
13500002	2	1	1	0	35
13500003	1	0	2	0	35
13500004	1	1	2	0	35
13500005	2	1	2	0	35
13500006	1	0	2	1	35
13500007	1	1	2	1	35
13500008	2	1	2	1	35

**FAME Users Define Cost & Performance Characteristics of Trade Options in Concise Catalog**

# Integration of Legacy Analyses

- LM CAUSE is represented in FAME as a black box entity to protect IP
- Inputs & outputs are captured as an approximation model connected to the UUV Concise Model
- FAME can also integrate legacy analyses as white / gray box models

## Input Combinations

<ul style="list-style-type: none"> <li>▪ <b>Payload</b> <ul style="list-style-type: none"> <li>▪ Size</li> <li>▪ Weight</li> <li>▪ Power</li> </ul> </li> <li>▪ <b>Mission</b> <ul style="list-style-type: none"> <li>▪ Range</li> <li>▪ Speed</li> <li>▪ Depth</li> <li>▪ Current</li> <li>▪ Salinity</li> </ul> </li> <li>▪ <b>Design Constraints</b></li> <li>▪ <b>Subsystem Configuration</b></li> </ul>
--



“Black Box”



## CSV inputs/outputs table(s)

<ul style="list-style-type: none"> <li>▪ <b>Payload</b> <ul style="list-style-type: none"> <li>▪ Size</li> <li>▪ Weight</li> <li>▪ Power</li> </ul> </li> <li>▪ <b>Mission</b> <ul style="list-style-type: none"> <li>▪ Range</li> <li>▪ Speed</li> <li>▪ Depth</li> <li>▪ Current</li> <li>▪ Salinity</li> </ul> </li> <li>▪ <b>Design Constraints</b></li> <li>▪ <b>Subsystem Configuration</b></li> </ul>	<ul style="list-style-type: none"> <li>▪ Vehicle Dimensions</li> <li>▪ Vehicle Weight</li> <li>▪ Subsystem Sizes</li> <li>▪ Subsystem Weights</li> <li>▪ Vehicle Reliability</li> <li>▪ Vehicle Cost</li> </ul>
--	---

**Black Box Integration Provides a Method to Protect Intellectual Property**

# Operational data integration

- Operational data provides the opportunity to enhance trade assessments with real measured cost and performance values for trade options
- Operational data is often in a format that does not readily support direct processing in trade metrics
- Challenge: how to integrate Operational Data with Data for Optimization?

**Operational data**

**«DataSet» ProcurementData**

Values

- partCost:double
- partDefectRate:double
- partID:int
- partLeadTime:double
- prodID:int
- SupplierID:int
- SupplierName:string
- suppQualityRating:double
- SystemID:int

**«DataSet» ProductionData**

Values

- assemblyDuration:double
- labor:double
- laborRate:double
- prodID:int
- subSystemID:int
- supportMaterialsCost:double
- systemID:int
- vehicleID:int

**«DataSet» MaintenanceLog**

Values

- downTime:double
- duration:double
- frequency:double
- maintenanceID:int
- maintenanceTime:d...
- PartID:int
- subSystemID:int
- vehicleID:int

**«DataSet» VehiclePerformance**

Values

- attribute\_0:int
- missionDate:double
- missionDesc:RhpString
- missionDuration:double
- missionID:double
- missionType:double
- vehicleInstID:double
- vehicleModel:double
- vehicleStability:double
- vehRecoverSuccessRate:double
- vehRecoveryTime:double

**«DataSet» PartFailureRecord**

Values

- faultDate:double
- faultID:int
- faultLevel:double
- missionID:int
- partID:int
- systemID:int
- vehicleID:int

- Large data set
- Different structure / sources / formats

How to fill the Manufacturing Labor Time attribute with measured Operational Data?

**Catalog data**

**Energy**

subSystemID	Cost	Labor Cost
1001	100	
1002	200	
1003	50	
1004	400	
1005	20	

**Propulsion**

subSystemID	Cost	Labor Cost
2001	80	
2002	70	
2003	78	
2004	50	
2005	70	

- Catalog data – small data set
- Input for the Optimization Engine



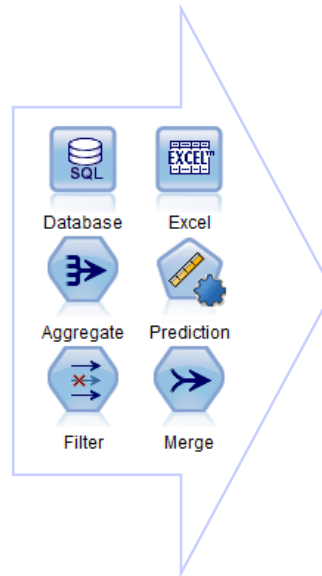
# Operational data integration

## Operational data

«DataSet» ProcurementData	«DataSet» ProductionData
Values	Values
partCost:double	assemblyDuration:double
partDefectRate:double	labor:double
partID:int	laborRate:double
partLeadTime:double	prodID:int
prodID:int	subSystemID:int
SupplierID:int	supportMaterialsCost:double
SupplierName:string	systemID:int
suppQualityRating:double	vehicleID:int
SystemID:int	

«DataSet» MaintenanceLog	«DataSet» VehiclePerformance	«DataSet» PartFailureRecord
Values	Values	Values
downTime:double	attribute_0:int	faultDate:double
duration:double	missionDesc:rlgString	faultID:int
frequency:double	missionDuration:double	faultLevel:double
maintenanceID:int	missionID:double	missionID:int
maintenanceTime:d...	missionType:double	partID:int
PartID:int	vehicleInstID:double	systemID:int
subSystemID:int	vehicleModel:double	vehicleID:int
vehicleID:int	vehicleStability:double	
	vehRecoverSuccessRate:double	
	vehRecoveryTime:double	

- Large data set
- Different structure / sources / formats



## Catalog data

### Energy

subSystemID	Cost	Labor Cost
1001	100	
1002	200	
1003	50	
1004	400	
1005	20	

### Propulsion

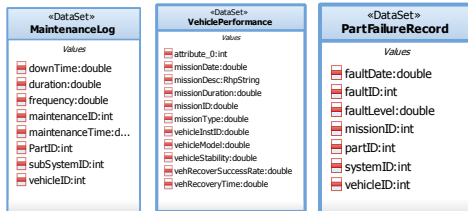
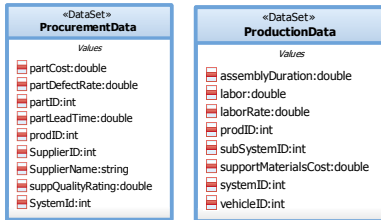
subSystemID	Cost	Labor Cost
2001	80	
2002	70	
2003	78	
2004	50	
2005	70	

- Catalog data – small data set
- Input for the Optimization Engine

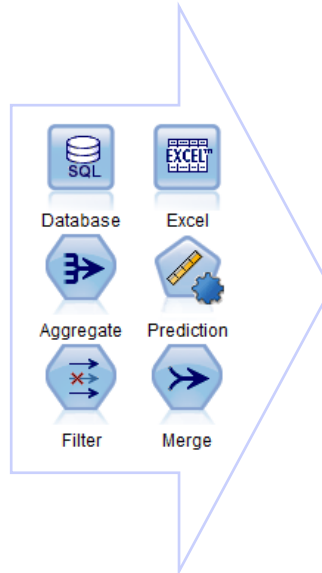
Use data flow  
definition & apply  
data transformation

# Operational data integration

## Operational data



- Large data set
- Different structure / sources / formats



## Catalog data

### Energy

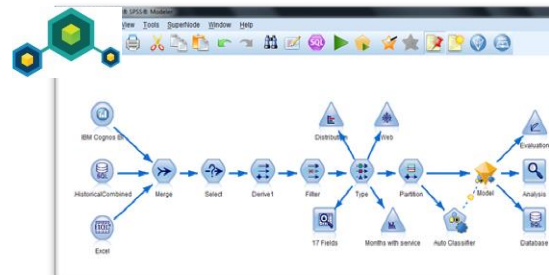
subSystemID	Cost	Labor Cost
1001	100	25
1002	200	30
1003	50	28
1004	400	17
1005	20	33

### Propulsion

subSystemID	Cost	Labor Cost
2001	80	14
2002	70	15
2003	78	13.5
2004	50	17
2005	70	12

- Catalog data – small data set
- Input for the Optimization Engine

## Solution: IBM SPSS Modeler



**\*NOTE: Numerical values are fictional, not actual, and intended only to demonstrate the functionality of FAME.**

# Operational Data Integration

## Catalog data for Propulsion

The Catalog ID

IDs to correlate with Cause data

id	IN_SUB_propulsc	IN_SUB_propulsor_ducte	IN_SUB_propulsor_redu	IN_SUB_propulsor_r	MaintenanceFi	MaintenancePlanCo	MaintenanceUnPlanCc	LogisticsTime	NumOfSpares	RepairTime	name
16100000	1	1	0	1	35	4000	6000	6000	1	24	Single_Duct_NoRed
16100001	1	0	1	2	35	6000	6000	4000	2	18	Double_Open_Redun

## CAUSE data for Propulsion

The CAUSE data ID

Input configuration

CAUSE output

Extract from the Manufacturing Operational Data

id	IN_SUB_propulsor_num	IN_SUB_propulsor_redundancy	IN_SUB_propulsor_type	IN_SUB_propulsor_ducted	OUT_propulsor_volume	OUT_Propulsor_Diameter	ConfigId	MC_mttf_effective	LaborTime	UnitCost
38000000	1	0	1	1	46623.73928	1.854019446	1	1414788.576		
38000001	1	0	1	1	25056.1019	1.507277846	2	1414788.576		
38000002	1	0	1	1	29362.26699	1.58912605	3	1414788.576		
38000003	2	1	1	0	30295.79777	2.538524122	4	1414788.576		
38000004	2	1	1	0	15009.18252	2.008299673	5	1414788.576		
38000005	2	1	1	0	17916.54636	2.130503092	6	1414788.576		
38000006	1	0	1	1	40343.11383	1.766702423	7	1414788.576		

## Manufacturing data

UUV production data modeled after SAP schema

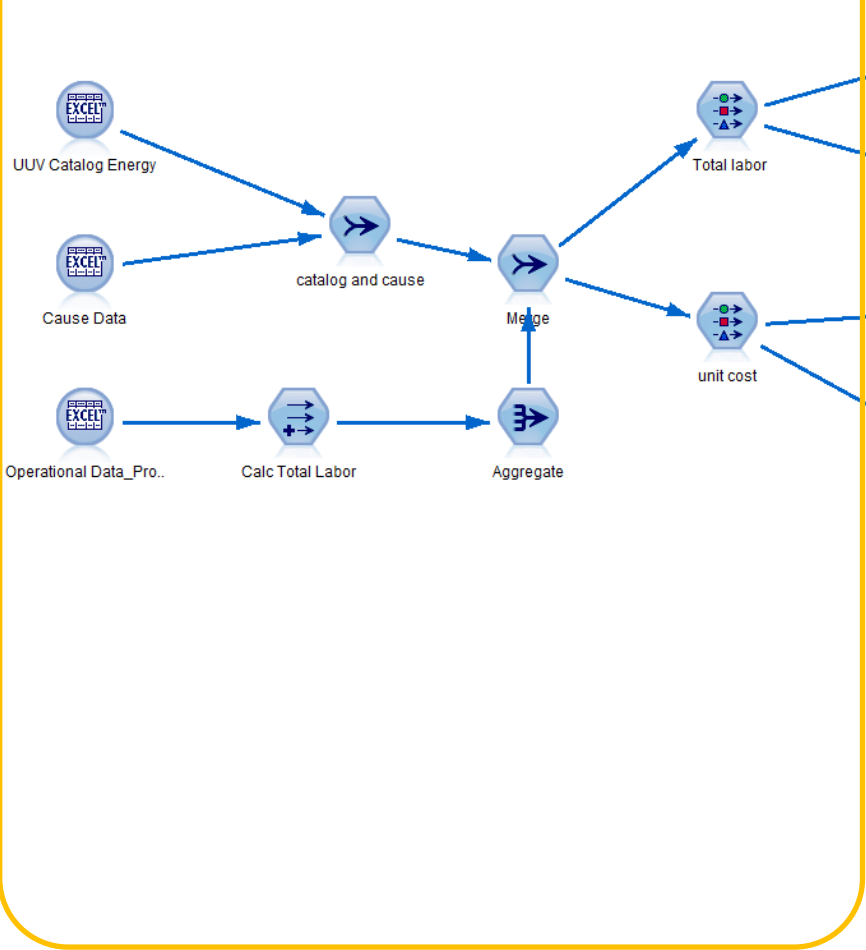
Additional data input supports mapping operational data to

Program ID	Part Name	PartNumber	Qty	UnitCost	Manufacturing Labor (hrs)	Procurement Labor (hrs)	RefVehVol (in^3)	RefVehkWh (kWhr)	Catalog Element	Catalog ID	Catalog Name

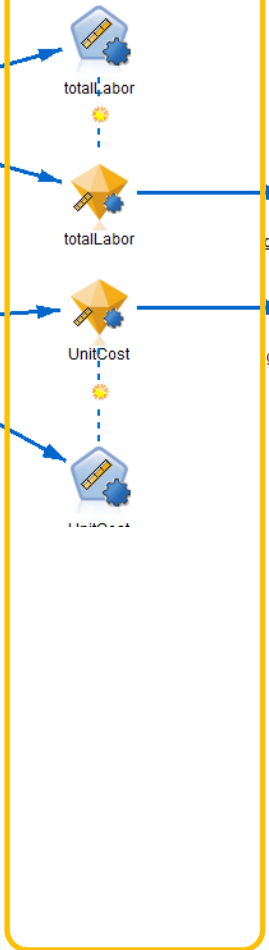
**\*NOTE: Numerical values are fictional, not actual, and intended only to demonstrate the functionality of FAME.**

# The SPSS Process for Energy

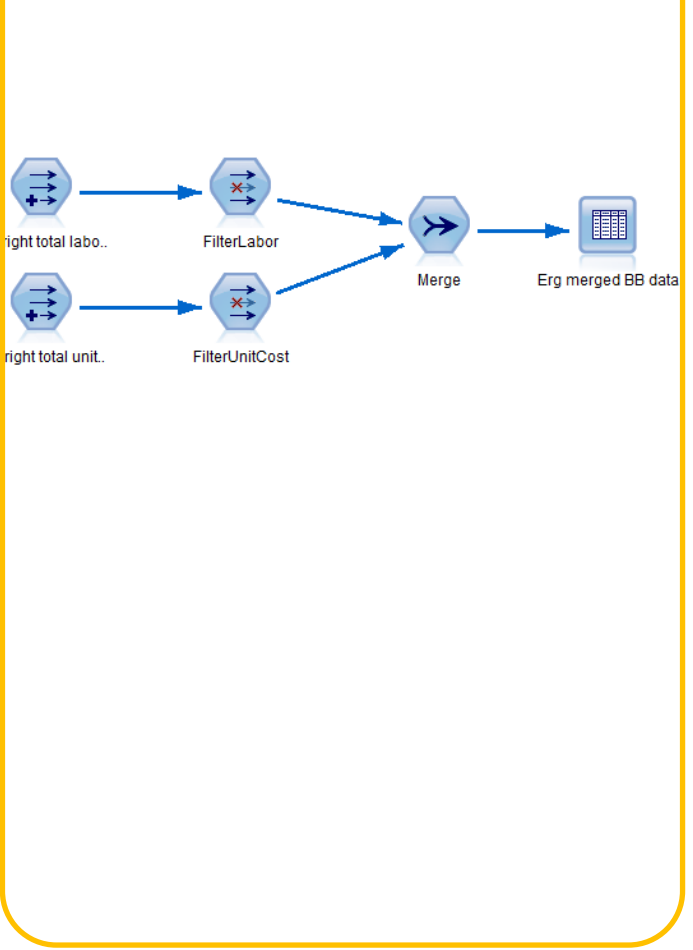
## Prepare the Data



## Prediction Models



## Output



# SPSS Results for Energy

ConfigId	id	name	UnitCost	totalLabor
1.000	4800000...	Engy Batt Secondary	121439...	21.818
16.000	4800000...	Engy Batt Secondary	121439...	21.818
34.000	4800000...	Engy Batt Secondary	121439...	21.818
13.000	4800000...	Engy Batt Secondary	121439...	21.818
4.000	4800000...	Engy Batt Secondary	121439...	21.818
31.000	4800000...	Engy Batt Secondary	121439...	21.818
22.000	4800000...	Engy Batt Secondary	121439...	21.818
7.000	4800000...	Engy Batt Secondary	121439...	21.818
28.000	4800000...	Engy Batt Secondary	121439...	21.818
19.000	4800000...	Engy Batt Secondary	121439...	21.818
10.000	4800000...	Engy Batt Secondary	121439...	21.818
25.000	4800000...	Engy Batt Secondary	121439...	21.818
18.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
33.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
30.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
27.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
24.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
21.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
36.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
15.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
12.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
9.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
6.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
3.000	4800002...	Engy Batt FutureSec	\$null\$	\$null\$
35.000	4800001...	Engy Batt Primary	337237...	23.000
11.000	4800001...	Engy Batt Primary	337237...	23.000
26.000	4800001...	Engy Batt Primary	337237...	23.000
17.000	4800001...	Engy Batt Primary	337237...	23.000
8.000	4800001...	Engy Batt Primary	337237...	23.000
29.000	4800001...	Engy Batt Primary	337237...	23.000
14.000	4800001...	Engy Batt Primary	337237...	23.000
5.000	4800001...	Engy Batt Primary	337237...	23.000
32.000	4800001...	Engy Batt Primary	337237...	23.000
23.000	4800001...	Engy Batt Primary	337237...	23.000
2.000	4800001...	Engy Batt Primary	337237...	23.000
20.000	4800001...	Engy Batt Primary	337237...	23.000

## Predictions

- Regression
- Generalized Linear
- KNN Algorithm
- SVM
- C&R Tree
- CHAID
- Neural Net
- Linear

ConfigId	id	name	totalLabor	UnitCost
1.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
16.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
34.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
13.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
4.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
31.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
22.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
7.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
28.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
19.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
10.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
25.000	4800000.0...	Engy Batt Secondary	21.818	121439.636
18.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
33.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
30.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
27.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
24.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
21.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
36.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
15.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
12.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
9.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
6.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
3.000	4800002.0...	Engy Batt FutureSec	21.818	121439.636
35.000	4800001.0...	Engy Batt Primary	23.000	337237.500
11.000	4800001.0...	Engy Batt Primary	23.000	337237.500
26.000	4800001.0...	Engy Batt Primary	23.000	337237.500
17.000	4800001.0...	Engy Batt Primary	23.000	337237.500
8.000	4800001.0...	Engy Batt Primary	23.000	337237.500
29.000	4800001.0...	Engy Batt Primary	23.000	337237.500
14.000	4800001.0...	Engy Batt Primary	23.000	337237.500
5.000	4800001.0...	Engy Batt Primary	23.000	337237.500
32.000	4800001.0...	Engy Batt Primary	23.000	337237.500
23.000	4800001.0...	Engy Batt Primary	23.000	337237.500
2.000	4800001.0...	Engy Batt Primary	23.000	337237.500
20.000	4800001.0...	Engy Batt Primary	23.000	337237.500

**CAUSE does not have cost prediction for Future Battery**

**SPSS prediction capability evaluates Future Battery Unit Cost**

# Agenda

- From SE to MBCE
- AOW background
- PORTALS
- FAME
- **EMI**
- DANSE
- Summary

# EMI: Engineering Management Integrator

- SE cycle vs Project Management Gantt
- Cooperation instead of twisting arms
- IBM Research – Technion collaboration (Prof. Avy Shtub, Michal Iluz)
- Integration of AOW with Project Team Builder (PTB)

# Time Management for Architecture Optimization

- Multi-mode Resource Constrained Project Scheduling Problem (MRCPSP)
  - project activities have several operational modes
  - each mode has its own duration and required set of resources
  - precedence constraints between activities
  - resources have a final capacity
  - solution defines the mode in which each activity is executed and schedules the activities
  
- Adjusting MRCPSP to AO
  - Synchronization of mode selection with architectural decisions
  - Part time job intensity
  - Variable period lengths
  - ...



# Mathematical formulation - model

AO-MRCPSP

$$\text{Minimize } \{C_{max}, D\} \quad (1)$$

Subject to

$$\sum_{j \in M_i} x_{ij} = 1 \quad \forall i \in A \quad (2)$$

$$y_{jt} \leq x_{ij} \quad \forall i \in A, j \in M_i, t \in P \quad (3)$$

$$w \cdot \tilde{y}_{it} \leq \sum_{j \in M_i} y_{jt} \quad \forall i \in A, t \in P \quad (4)$$

$$\tilde{y}_{it} \geq \sum_{j \in M_i} y_{jt} \quad \forall i \in A, t \in P \quad (5)$$

$$s_{it} \leq s_{i,t+1} \quad \forall i \in A, t \in P | t < T \quad (6)$$

$$f_{it} \leq f_{i,t+1} \quad \forall i \in A, t \in P | t < T \quad (7)$$

$$s_{it} \geq \tilde{y}_{it} \quad \forall i \in A, t \in P \quad (8)$$

$$f_{it} \leq 1 - \tilde{y}_{it} \quad \forall i \in A, t \in P \quad (9)$$

$$\tilde{y}_{it} \leq f_{i't} \quad \forall i \in A, i' \in IP_i, t \in P \quad (10)$$

$$\sum_{j \in M_i} \sum_{t \in P} y_{jt} p_t \geq \sum_{j \in M_i} x_{ij} d_j \quad \forall i \in A \quad (11)$$

$$\sum_{t \in P} (s_{it} - f_{it}) p_t \leq e \sum_{j \in M_i} x_{ij} d_j \quad \forall i \in A \quad (12)$$

$$C_i = T_{max} - \sum_{t \in P} f_t p_t \quad \forall i \in A \quad (13)$$

$$C_{max} \geq C_i \quad \forall i \in A \quad (14)$$

$$u_{kt} = \frac{1}{v_k} \sum_{i \in A} \sum_{j \in M_i} y_{jt} r_{jk} \quad \forall k \in R, t \in P \quad (15)$$

$$u_{kt} \leq 1 \quad \forall k \in R, t \in P \quad (16)$$

$$u_k = \frac{\sum_{t \in P} u_{kt} p_t}{\sum_{t \in P} p_t} \quad (17)$$

$$B = \sum_{k \in R} u_k v_k b_k \quad (18)$$

$$A = \sum_{i \in A} \sum_{j \in M_i} x_{ij} a_j \quad (19)$$

$$D = A + B \quad (20)$$

$$\sum_{j \in M_i} x_{ij} n_j = q_{h_i} \quad \forall i \in A \quad (21)$$

$$x_{ij}, \tilde{y}_{it}, s_{it}, f_{it} \in \{0,1\} \quad 0 \leq y_{jt}, u_{kt}, u_k \leq 1 \quad C_i, C_{max}, A, B, D \geq 0 \quad (22)$$

# Mathematical formulation - AOW

Minimize *Original objectives, Objectives (1)* (23)

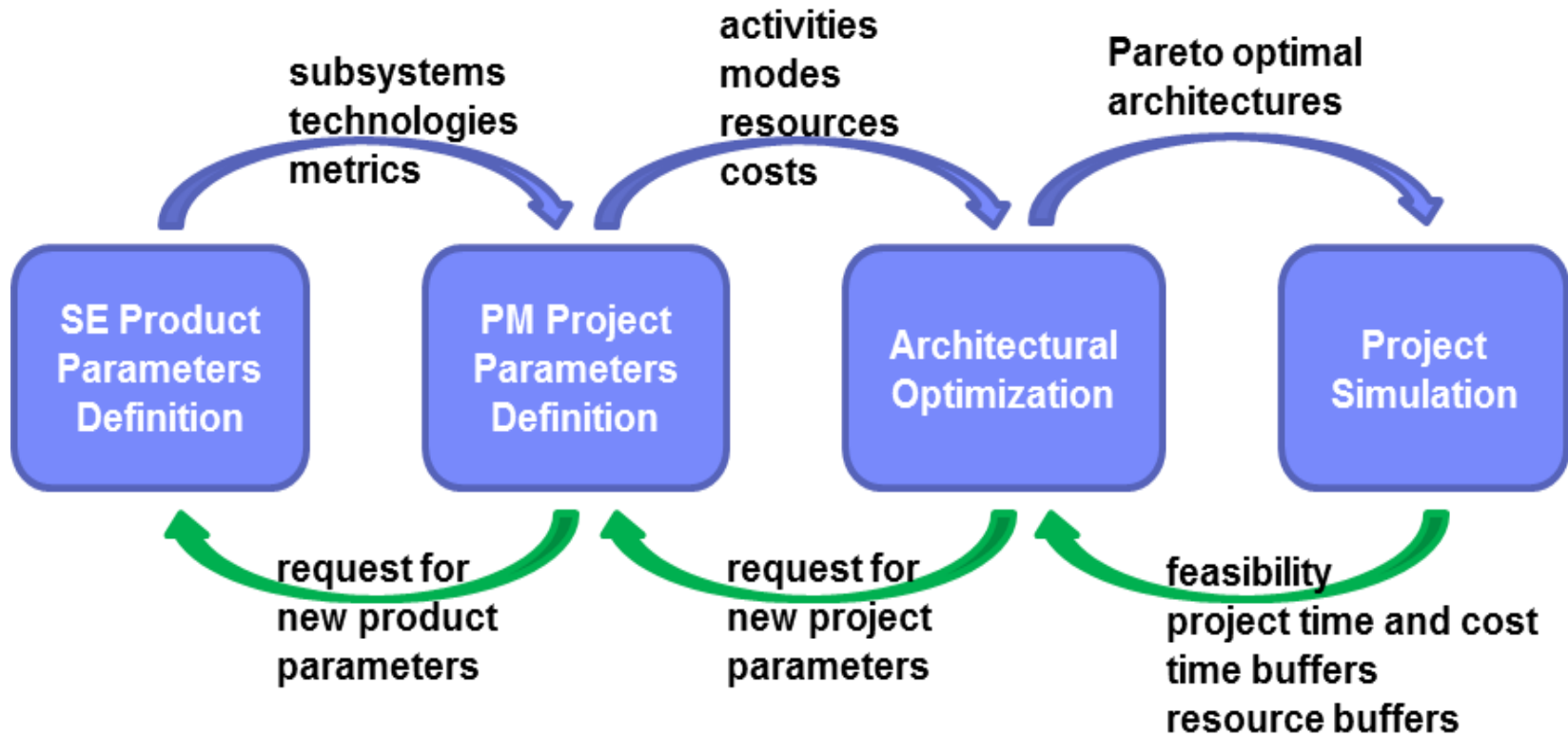
Subject to

*Original architectural constraints* (24)

Constraints (2) - (22)

*Subsystem/component type synchronization constraints* (25)

# EMI Process



# Airbus Group, Doors Management System

## Development and Analysis of a Simplified Doors Control System

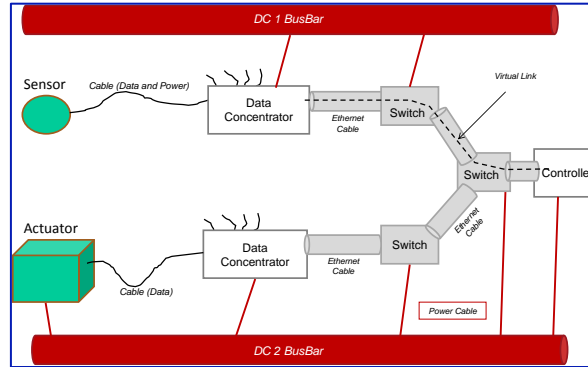


- Monitor and Control Passenger Doors, Emergency Exits, and Cargo Doors
- Design a system out of existing components for best weight, cost, power etc.

# Development and Analysis of a Doors and Slides Control System

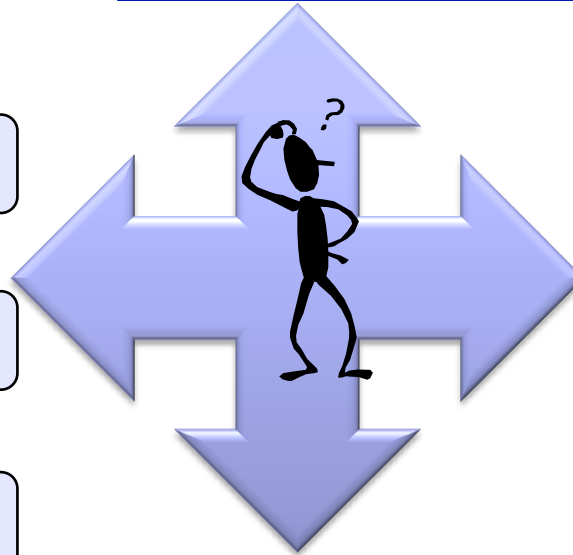
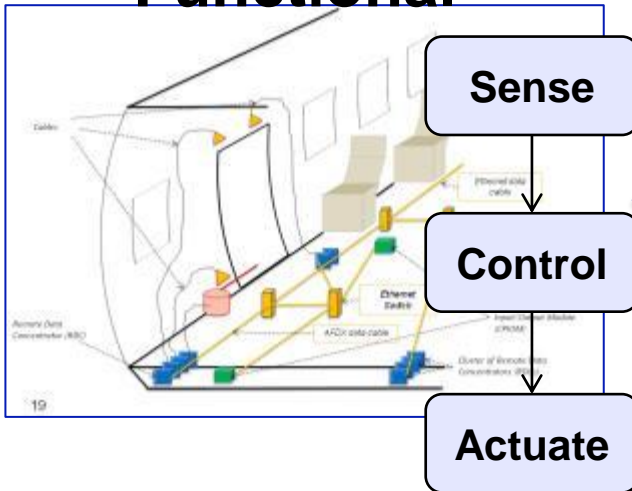


• Passenger Doors, Emergency Exits, and Cargo Doors



## Structure

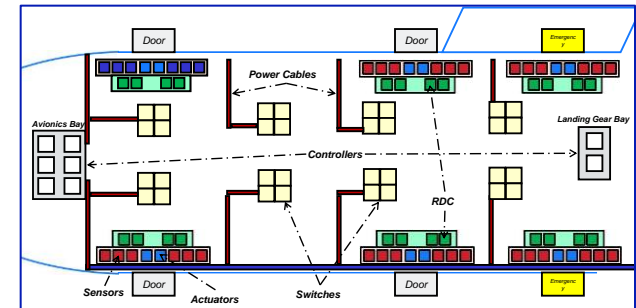
## Functional



## Design metrics

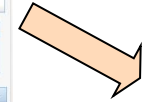
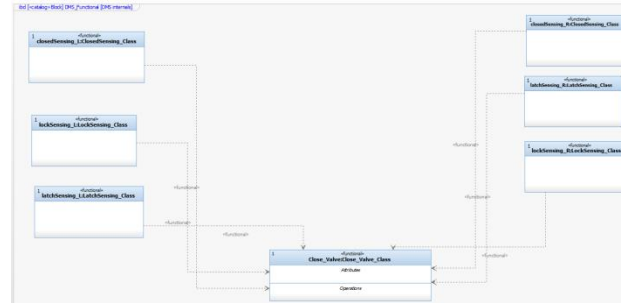
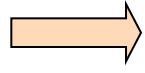
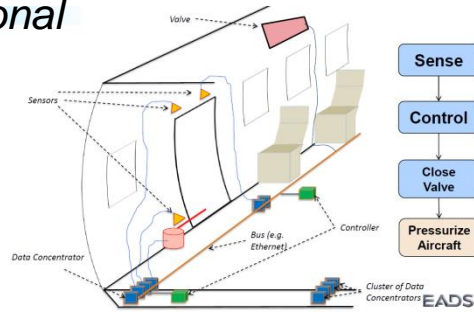
Weight, Cable length, Power distribution, Mapping, Allocation, Reliability ...

## Geometrical

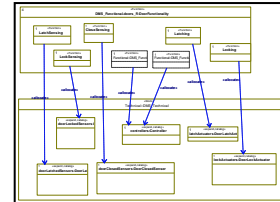


# DMS: From user story to model

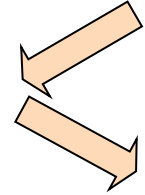
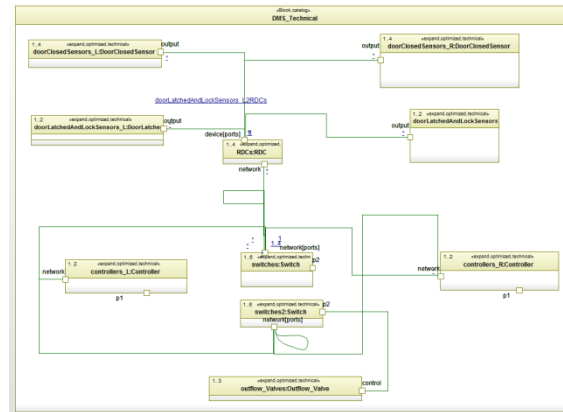
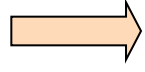
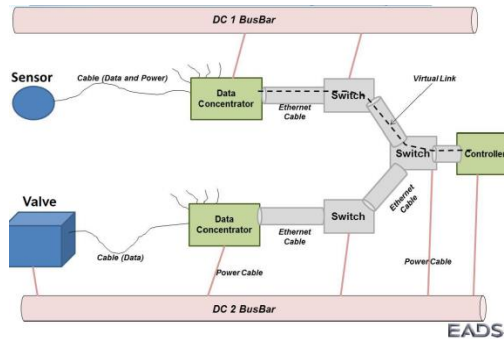
## Functional



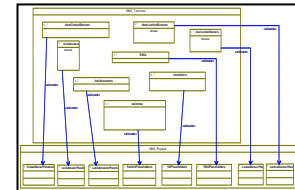
## Mapping



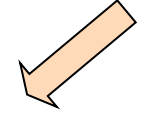
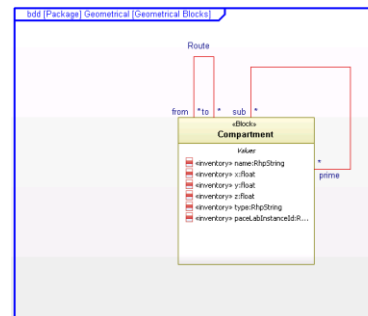
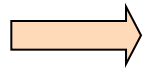
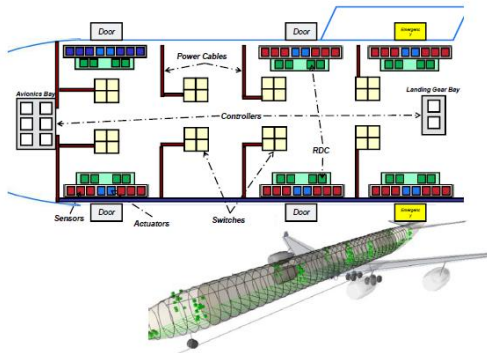
## Technical



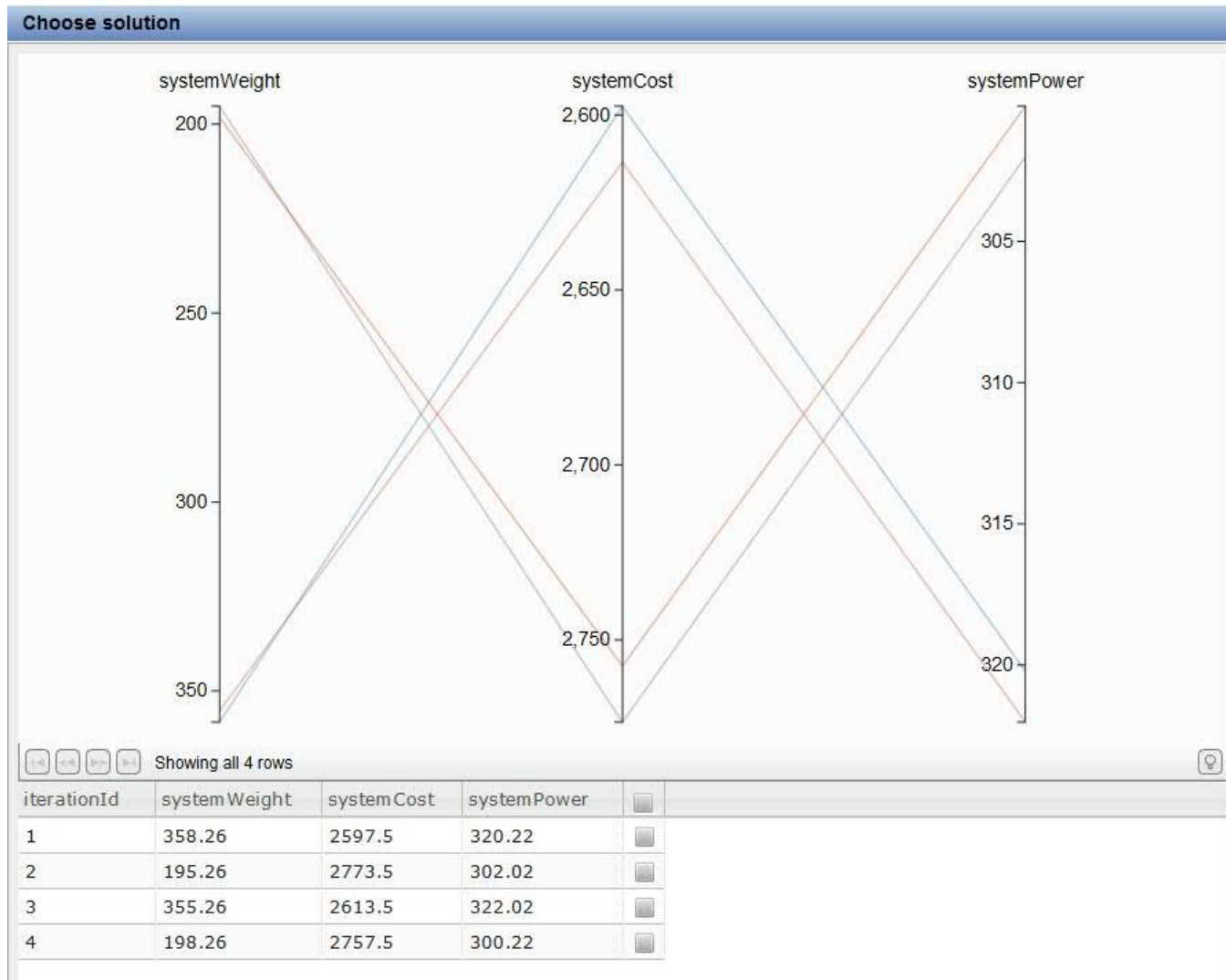
## Allocation



## Geometrical

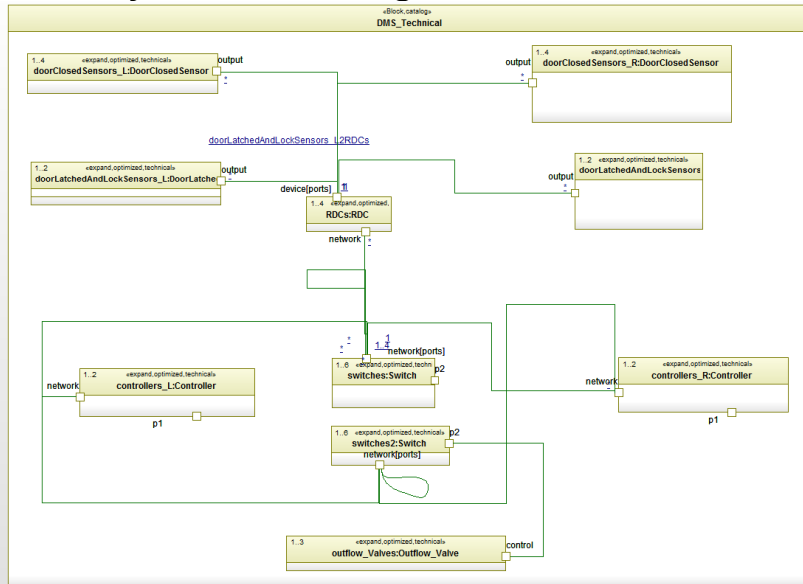


# AOW – AO



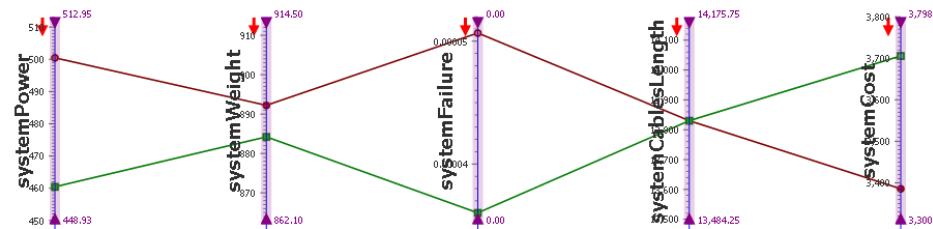
# DMS: Results [1/2]

## 1. SysML modeling

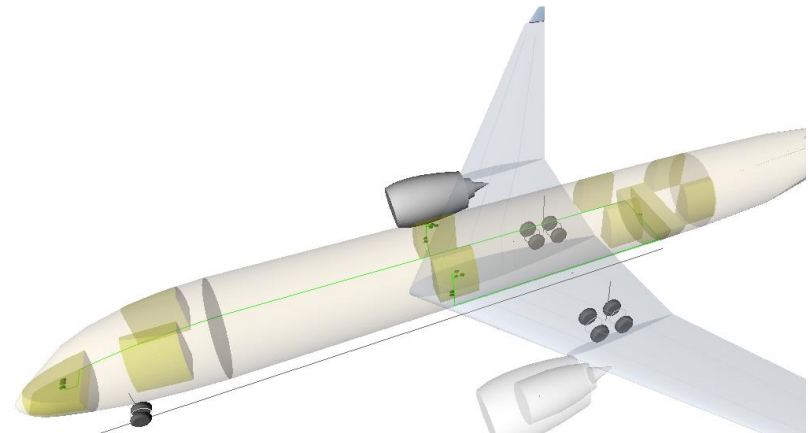
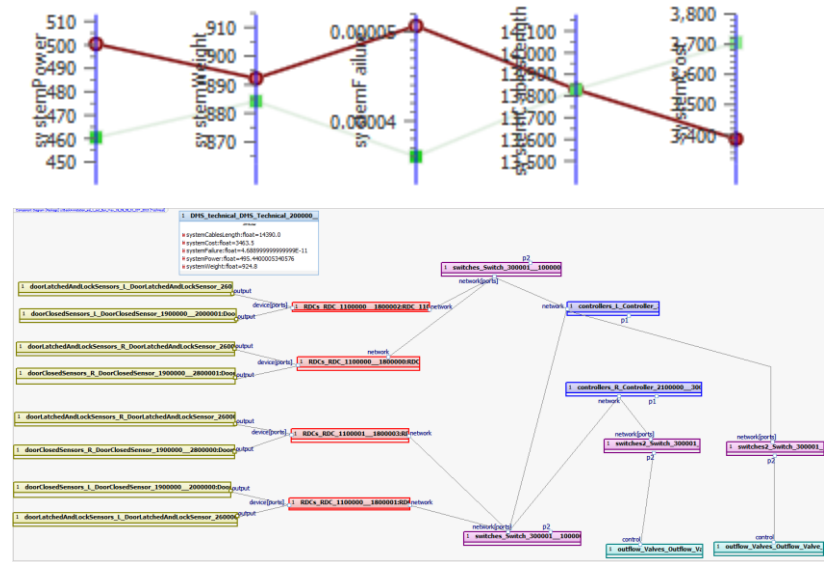


## 2. Run Optimization and show alternatives

ID	systemPower	systemWeight	systemFailure	systemCables.Length	systemCost	Finished at	Duration
2e0e937c-a4b9-4ac8-b0ab-702de3a1c94a	500.44	892.20	5.066E-5	13830.00	3385.50	May 23, 2013 11:14:54 AM	942 sec
3232cbf2-0383-4500-811a-52d643079ed3	460.44	884.20	3.602E-5	13830.00	3705.50	May 23, 2013 11:10:54 AM	938 sec



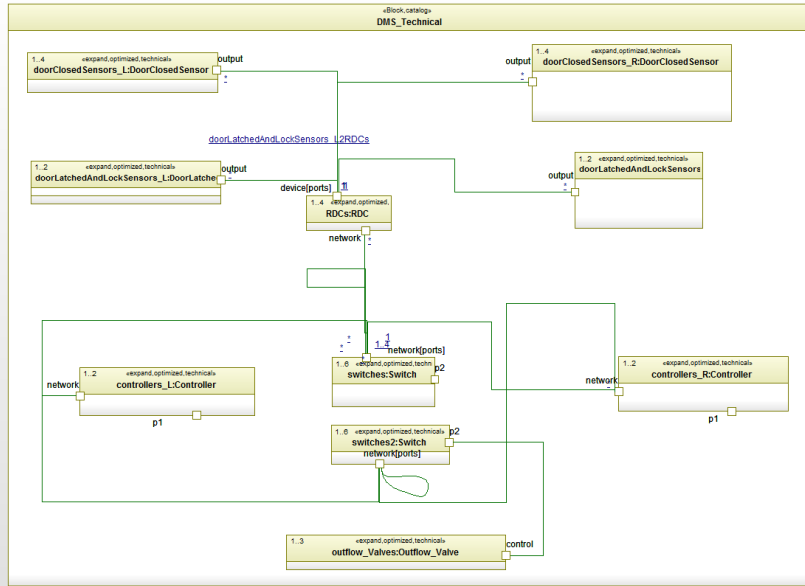
## 3. Visualize the alternatives





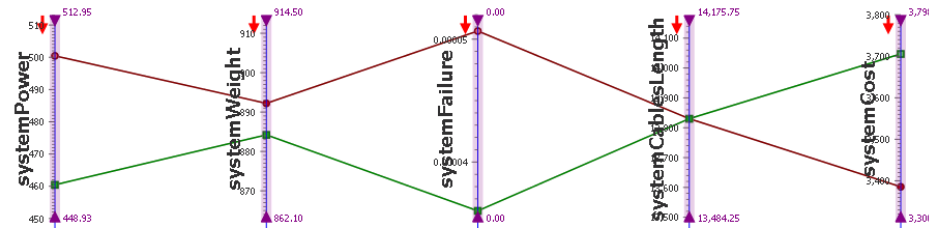
# DMS: Results [2/2]

## 1. SysML modeling

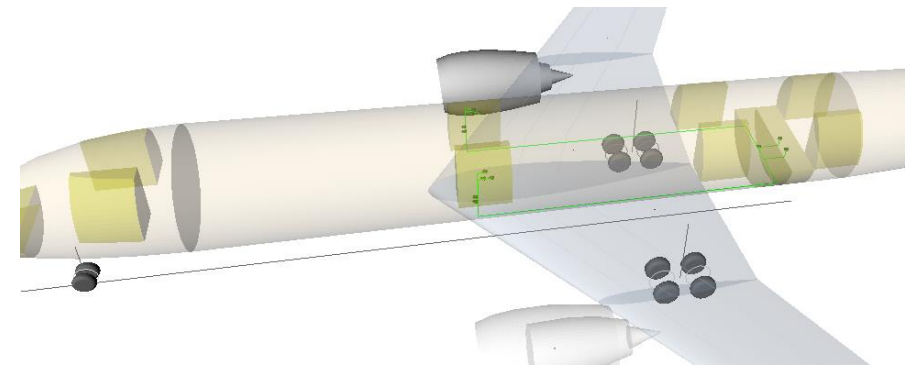
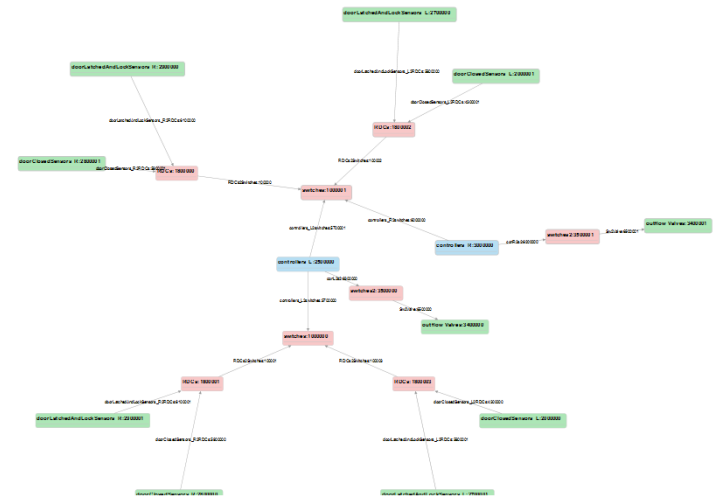
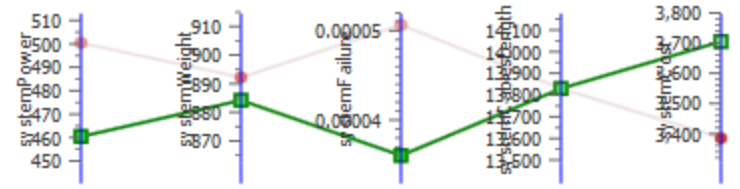


## 2. Run Optimization and show alternatives

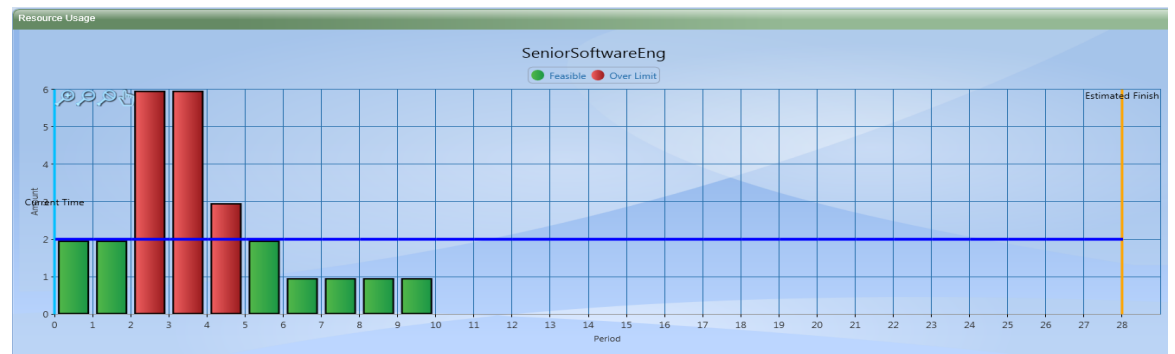
ID	systemPower	systemWeight	systemFailure	systemCables.Length	systemCost	Finished at	Duration
2e0e937c-a4b9-4ac8-b0ab-702de3a1c94a	500.44	892.20	5.066E-5	13830.00	3385.50	May 23, 2013 11:14:54 AM	942 sec
3232cbf2-0383-4500-811a-52d643079ed3	460.44	884.20	3.602E-5	13830.00	3705.50	May 23, 2013 11:10:54 AM	938 sec



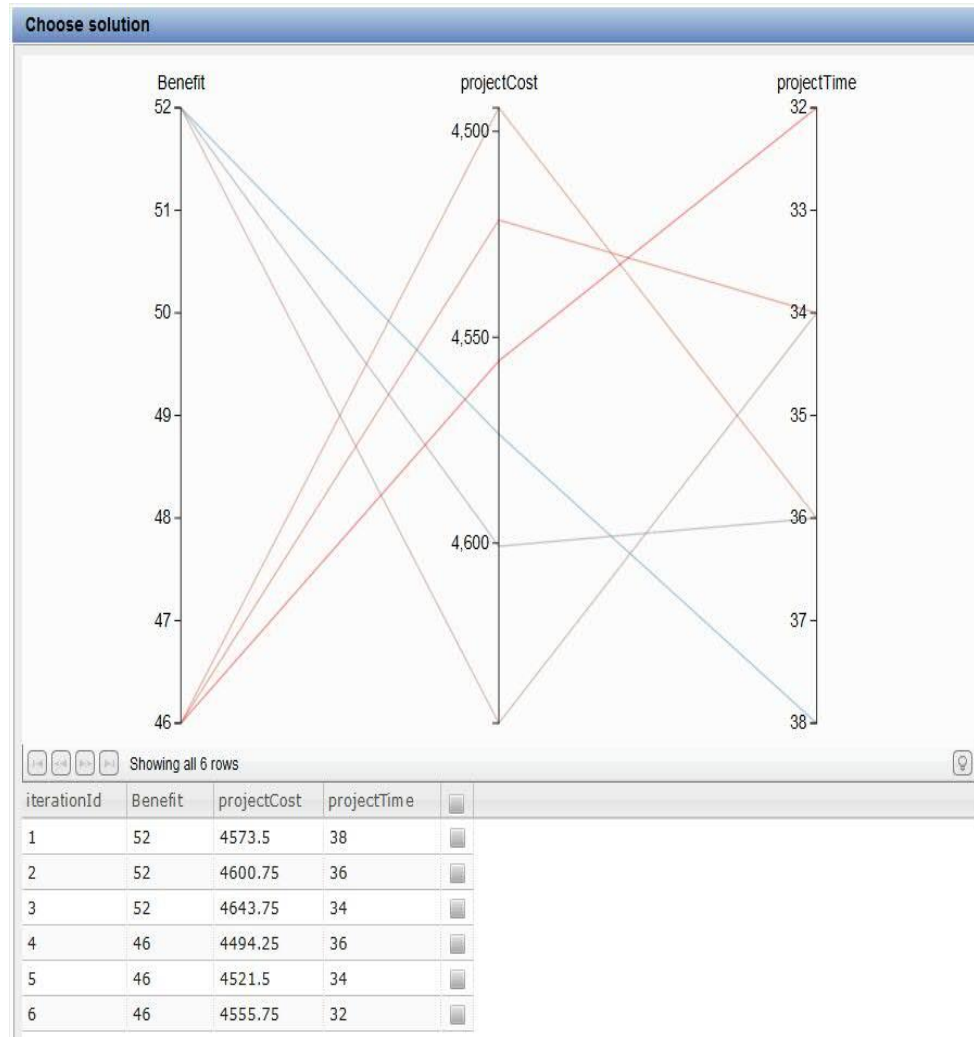
## 3. Visualize the alternatives



# Project Team Builder (PTB) model

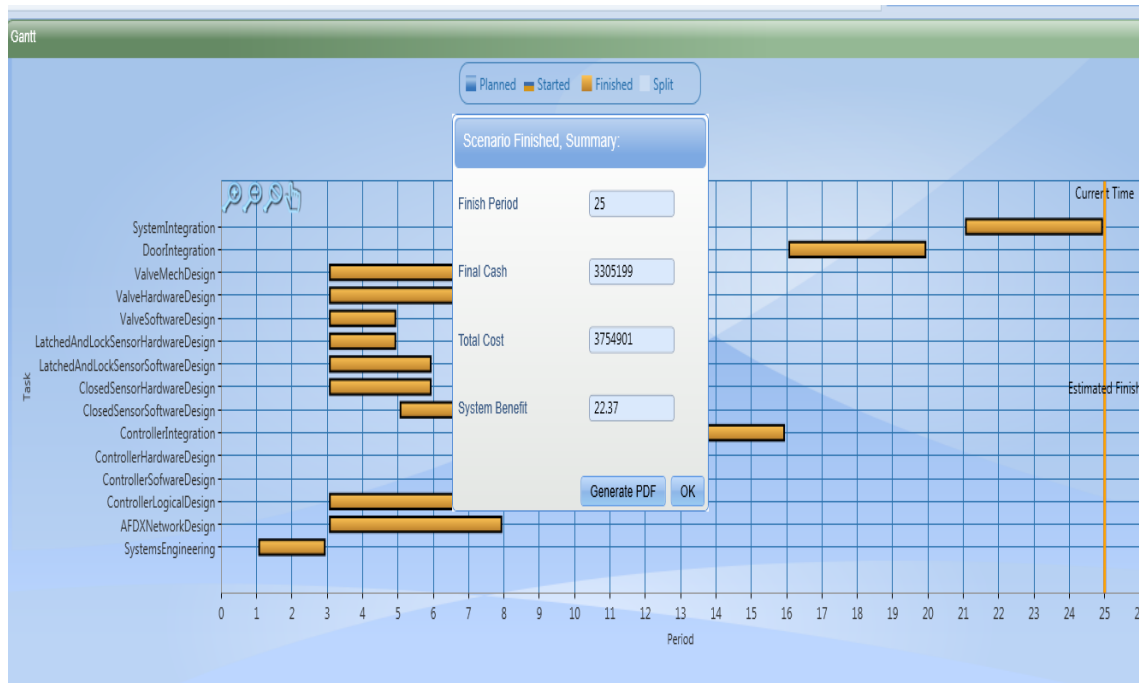


# AOW – PM



# AOW: AO-PM

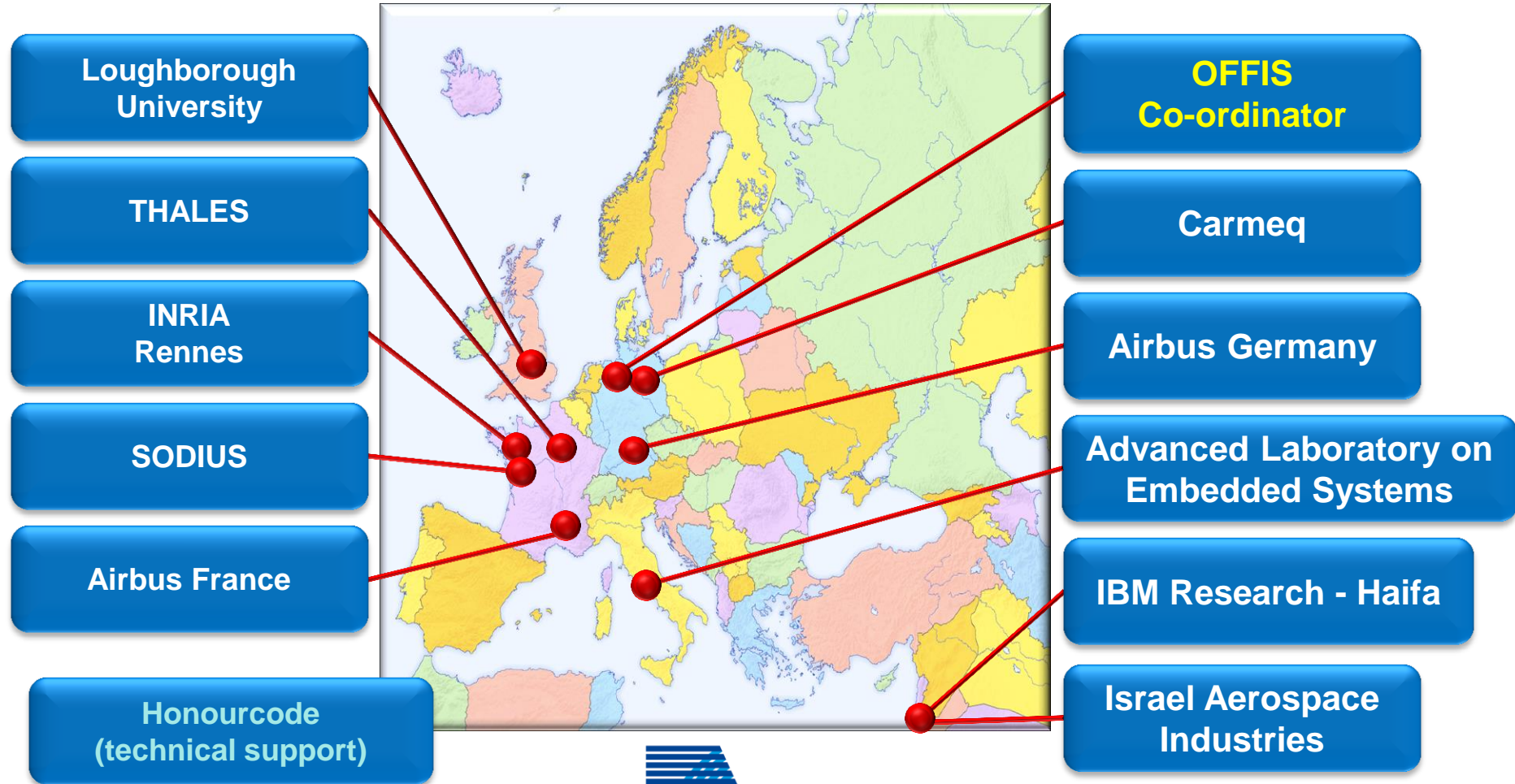
iterationI...	systemW...	systemC...	systemP...	projectTi...	projectC...	
1	195.3	2775.5	302.02	46	4662.5	
2	195.3	2775.5	302.02	48	4635.25	



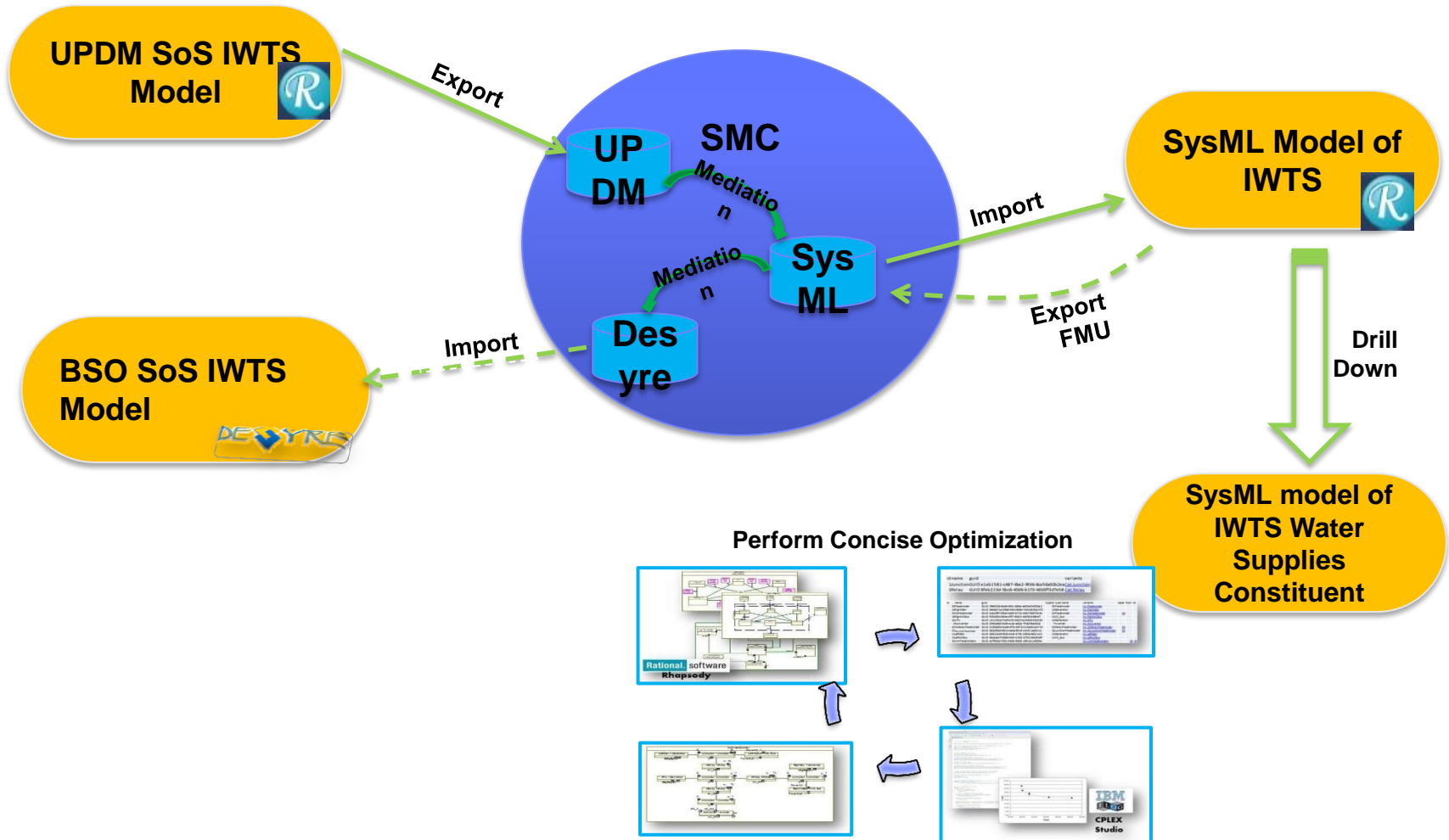
# Agenda

- From SE to MBCE
- AOW background
- PORTALS
- FAME
- EMI
- **DANSE**
- Summary

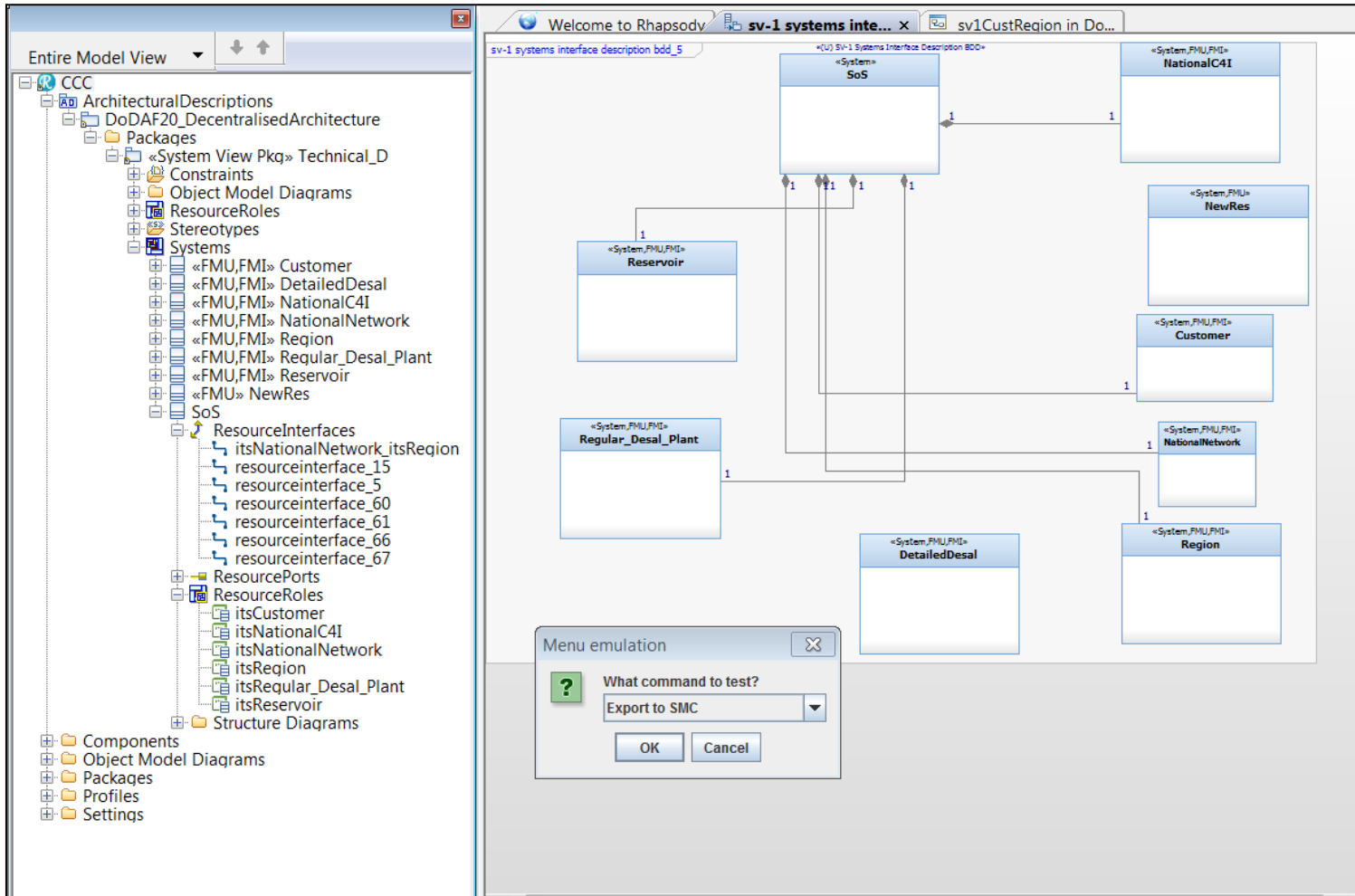
# DANSE: Design for Adaptability and evolutionN in System of systems Engineering



# Scenario Overview



# Full model (UPDM)



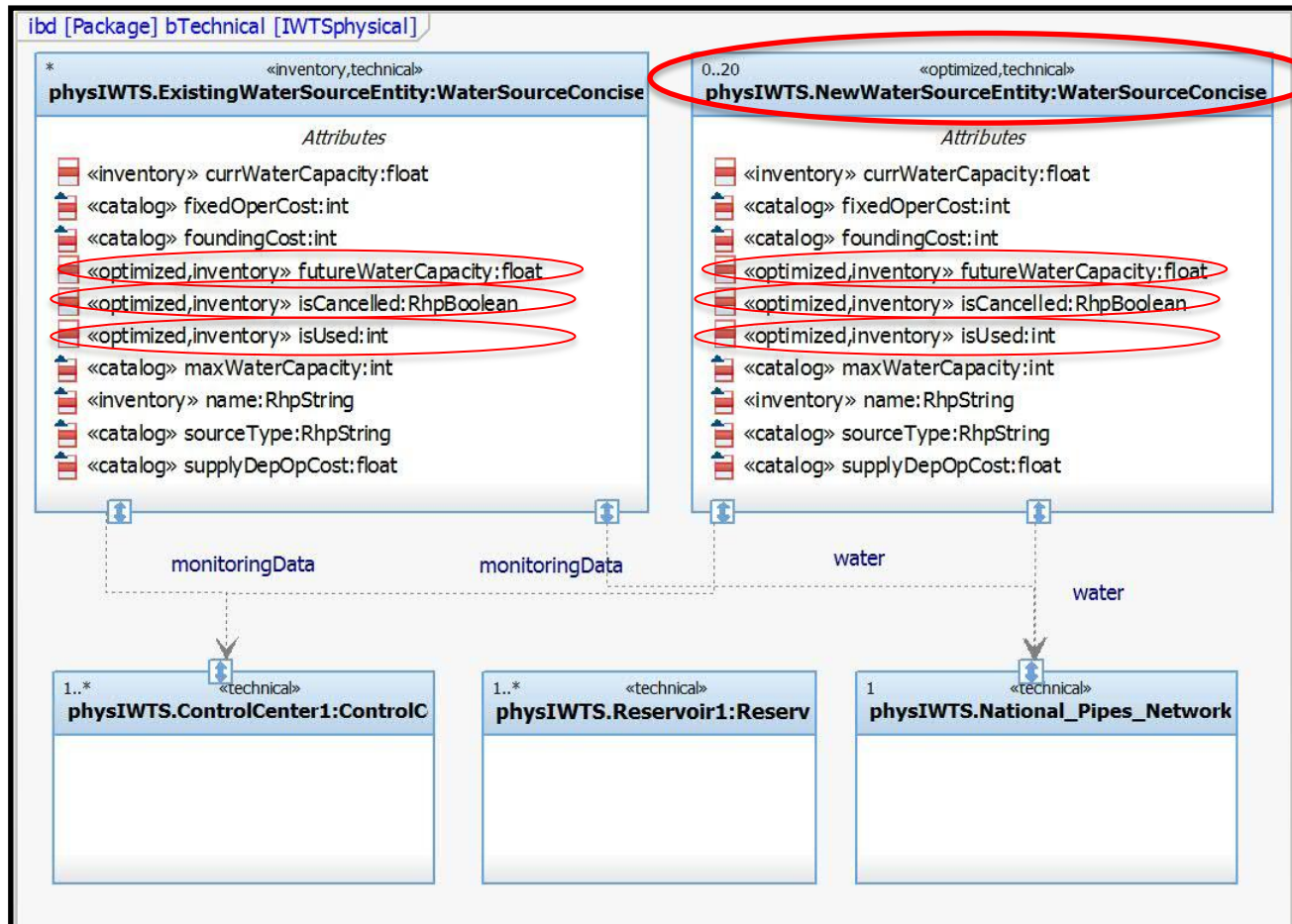


# Importing to Rhapsody SysML project

**Import, and populate a BDD automatically.**

The screenshot illustrates the process of importing a SysML project into Rhapsody. On the left, the 'Entire Model View' shows the project structure, including 'SysML\_CCC', 'Components', 'Internal Block Diagrams', 'Packages', and 'Profiles'. A 'Menu emulation' dialog box is open, asking 'What command to test?' with 'Import from SMC' selected. Below this, the 'Library (REF)' view shows a tree structure of packages and blocks, including 'Technical\_D' and 'SoS'. On the right, the 'Structure1' view displays a 'DoDAF20\_DecentralisedArchitecture' diagram. This diagram shows a 'Technical\_D' package containing several blocks: 'DetailedDesal' and 'NewRes' (both «Block,FMU,FMI»), and 'NationalC4I', 'Region', 'Customer', and 'Regular\_Desal\_Plant' (all «Block,FMI,FMI»). These blocks are connected to a central 'SoS' block («Block») via arrows, indicating dependencies. A note block is also present below the SoS block.

# Decision variables



# Excel tables (inventory & catalog)

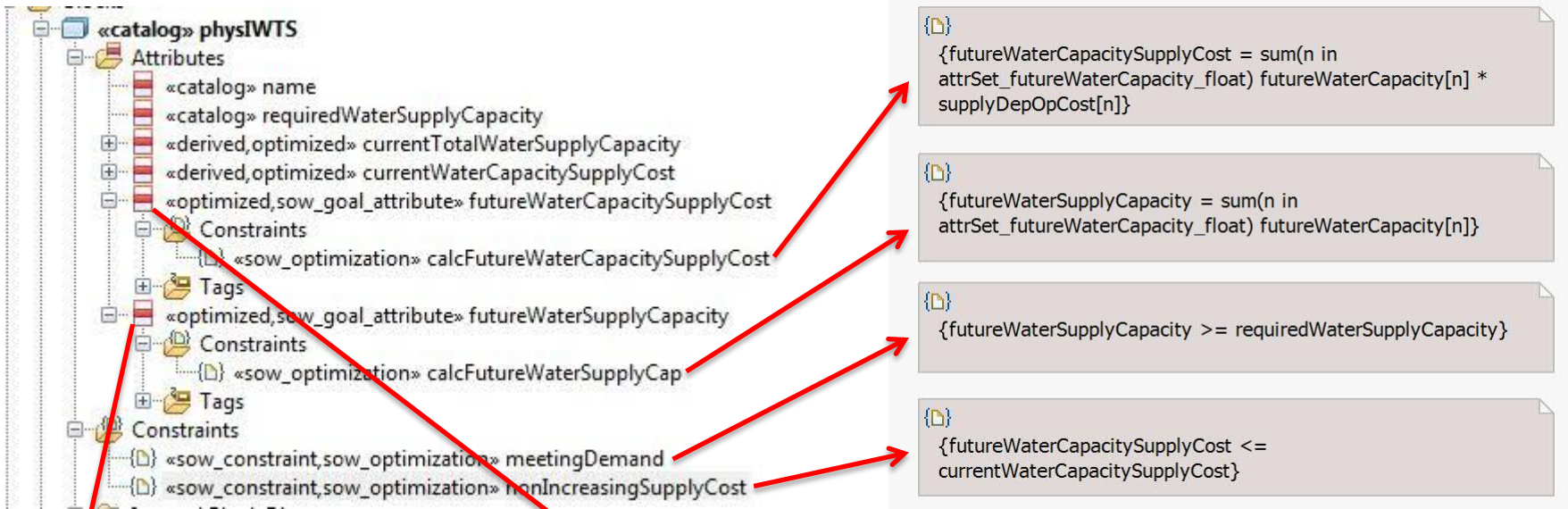
	A	B	C	D	E	F	G
1	int	int	int	int	float	RhpString	
2	id	maxWaterCapacity	foundingCost	fixedOperCost	supplyDepOpCost	sourceType	
3	1000000	10000			300	Desal_Large	
4	1000001	5000			1000	Desal_Medium	
5	1000002	6000			300	Ground_Water_Medium	
6	1000003	2500			300	Ground_Water_Small	
7	1000004	100000			100	Grand_National_Lake	
8							

Types Instances c\_physIWTS#9 c\_WaterSourceConcise#10

	A	B	C	D	E	F	G	H	I
1	int	int	int	int	float	float	RhpBoolean	RhpString	
2	id	owningPart	c_id	isUsed	futureWaterCapacity	currWaterCapacity	isCancelled	name	
3	1700000		1000000			2000		DesalL1	
4	1700001		1000000			2000		DesalL2	
5	1700002		1000001			1000		DesalM3	
6	1700003		1000001			2500		DesalM4	
7	1700004		1000001			5000		DesalM5	
8	1700005		1000001			500		DesalM6	
9	1700006		1000002			700		GroundM1	
10	1700007		1000002			700		GroundM2	
11	1700008		1000003			200		GroundS3	
12	1700009		1000003			2500		GroundS4	
13	1700010		1000004			70000		GrandLake1	
14									

c\_WaterSourceConcise#10 i\_ExistingWaterSourceEntity#17

# Goals and constraints



Attribute : futureWaterSupplyCapacity in physIWTS

Concise	
sow_goal_attribute	
action	maximize
description	
isEnabled	<input checked="" type="checkbox"/>
priority	1

Attribute : futureWaterCapacitySupplyCost in physIWTS

Concise	
sow_goal_attribute	
action	minimize
description	
isEnabled	<input checked="" type="checkbox"/>
priority	1E-3

0..20 «optimized,technical»  
**NewWaterSourceEntity:WaterSourceConcise**

Attributes

- «inventory» currWaterCapacity:float
- «catalog» fixedOperCost:int
- «catalog» foundingCost:int
- «optimized,inventory» futureWaterCapacity:f...
- «optimized,inventory» isCancelled:RhpBoolean
- «optimized,inventory» isUsed:int
- «catalog» maxWaterCapacity:int
- «inventory» name:RhpString
- «catalog» sourceType:RhpString
- «catalog» supplyDepOpCost:float

# Setting optimization parameters

**Run Optimization**

ID: 874d9221-6d9d-4d4c-8d94-026eba2fa2d6

Load Model from file: D:\Work\IDANSE\water\IWTS\concise\_new\IWTS\concise.xls

Merge Destination with external sources:

Loading .mod template from file: D:\DM4.0.5\_WS\com.ibm.sow.conciseplugin\res\lopt\_gen\_Nov2014.tmod

Loading .dat template from file: D:\DM4.0.5\_WS\com.ibm.sow.conciseplugin\res\lopt\_gen\_Nov2014.tdat

Saving .mod to file: D:\Work\IDANSE\water\IWTS\concise\_new\IWTS1.mod

Saving .dat to file: D:\Work\IDANSE\water\IWTS\concise\_new\IWTS1.dat

Run Optimization:

Optimization iterations limit: 5

Optimization alpha treshold: 0.0001

Back-annotate optimization results:

Optimization Run Time per iteration(sec): 360

Start Time for polishing (sec): 60

Memory Limit (Mb): 2,000

Relative optimality gap: 0

OK Cancel

# Running optimization

**DMA iteration number 4**

100%

Optimization run properties | Solver info | **Output log** | Error log

```

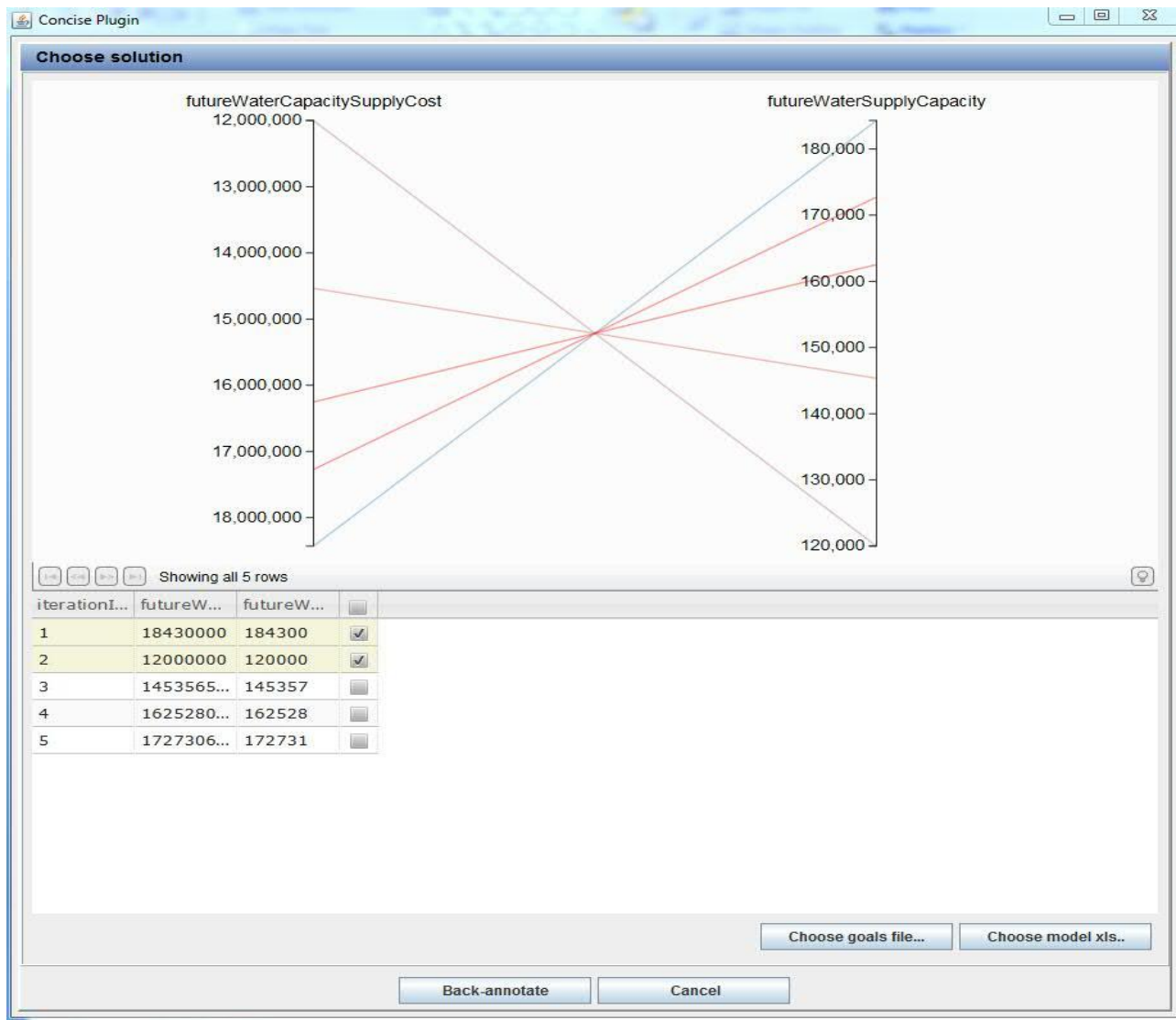
Gomory fractional cuts applied: 2

Root node processing (before b&c):
Real time      = 0.11 sec. (15.33 ticks)
Parallel b&c, 8 threads:
Real time      = 0.02 sec. (0.10 ticks)
Sync time (average) = 0.00 sec.
Wait time (average) = 0.00 sec.
-----
Total (root+branch&cut) = 0.13 sec. (15.43 ticks)
Card(setOf_ExistingWaterSourceEntity_17) = 11
Card(setOf_itsPhysIWTS_48) = 1
Card(setOf_Reservoir1_38) = 1
Card(setOf_ControlCenter1_22) = 1
Card(setOf_NewWaterSourceEntity_24) = 100
Card(setOf_National_Pipes_Network_30) = 1
Card(tcons) = 0
Card(nodes) = 115
Card(physNodesIndex) = 35
Card(parts) = 150

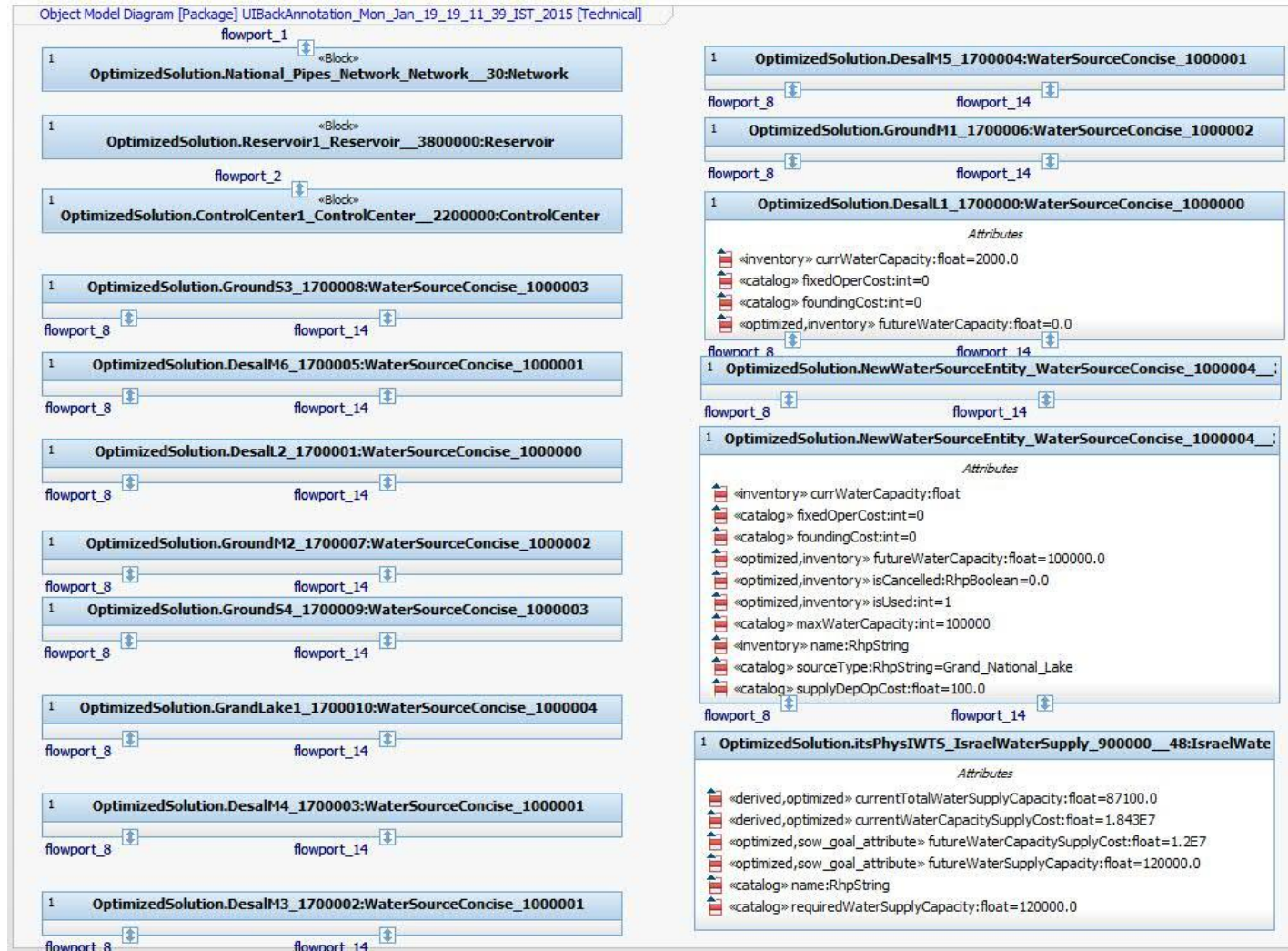
Card(attrSet_partStereotype_Interface) = 0
    
```

Abort | Finish iteration and stop | **Start optimization** | Show optimal solutions | Close

# Snake diagram



# One of optimal solutions





# Agenda

- From SE to MBCE
- AOW background
- PORTALS
- FAME
- EMI
- DANSE
- **Summary**

# Summary

- New paradigm – Model Based Continuous Engineering
- Model generation using free and restricted input
  - support from reusable Engineering Knowledge Base is mandatory
  - reusable structural and analysis libraries
- Open World Assumption
  - Power of incompleteness – mediate what is common
  - Property based semantic middleware (SEMI)
  - Integration of external data sources, legacy tools, and operational / lifecycle data
- Ontology based concepts and property-defined sets enable building reusable libraries for lifecycle optimization and analysis
  - Current reusable libraries: mapping, reliability, resource allocation, interface compatibility, power distribution, voltage drop, data flow, total availability, lifecycle cost, project management... (ongoing extension)



IBM Research – Haifa Labs

Thank you

Questions?



# Systems & IoT Engineering group

